Homecomputer

TI99/4A Collection

DIVE-BOMBER

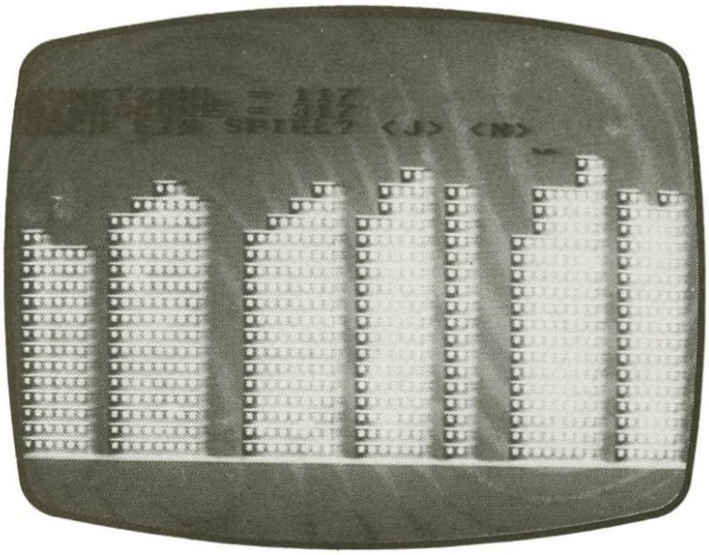
Ein Spielprogramm für den TI-99/4 in TI-BASIC.

Bei Dive-Bomber soll ein Flugzeug so lange wie möglich über eine Reihe Hochhäuser fliegen, ohne diesen zu nahe zu kommen. Da das Flugzeug laufend tiefer sinkt, kann dies nur durch gezielten Abwurf von Bomben verhindert werden, wodurch die Häuser Stockwerk um Stockwerk eliminiert werden.

Da der TI-99 in siner Grundversion keine vernünftigen Druckbefehle kennt, wurde eine Routine angehängt, die, wenn auch äußerst langsam, Texte an eine beliebige Stelle auf den Bildschirm schreibt.

Das Programm beginnt mit dem Initialisieren der RND-Funktion und dem Neu-Definieren einiger Zeichen und Farben. Danach werden die Häuser per Zufallszahlengenerator aufgebaut und das Flugzeug startet an oberen linken Bildrand. Durch Drücken einer Taste können Bomben abgeworfen werden, die die Stockwerke vom Einschlagpunkt aus nach oben löschen.

Wird vom Programm erkannt, daß das Flugzeug ein Haus streifen wird, ohne daß dies vom Spieler verhindert werden kann, so wird das Spiel abgebrochen und akustisch angezeigt, ob die alte Bestmarke (Highscore) übertroffen wurde.

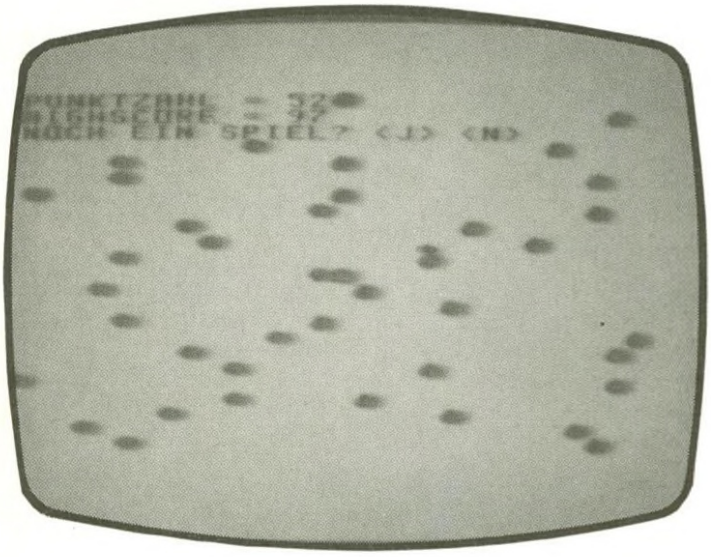


```
100 RANDOMIZE
110 CALL CLEAR
120 CALL CHAR(96,"00FF999999
FFFFFF")
130 CALL CHAR(112,"000000000
086CFFE")
140 CALL CHAR(104,"FFFFFF")
150 CALL CHAR(42,"0000006078
06")
160 CALL COLOR(9,5,16)
170 CALL COLOR(11,2,1)
180 CALL COLOR(10,15,1)
190 REM START NEUES SPIEL
200 PZ=0
210 REM HAEUSER AUFBAUEN
220 CALL CLEAR
230 I=2
240 J=RND*8+3
250 FOR B=1 TO RND*5
260 CALL VCHAR(J,I,96,24-J)
270 I=I+1
280 IF I>31 THEN 320
290 NEXT B
300 I=I+1
310 IF I<32 THEN 240
320 CALL HCHAR(24,1,104,32)
330 REM START BOMBER
340 FX=1
350 FY=1
360 LX=0
370 LY=1
380 LX=LX+1
390 IF LX<32 THEN 420
400 LY=LY+1
410 LX=1
420 CALL SOUND(-1000,-6,24-F
Y)
430 FX=FX+1
440 IF FX<32 THEN 470
450 FX=1
460 FY=FY+1
470 CALL GCHAR(FY,FX,T)
480 IF T=96 THEN 800
490 CALL GCHAR(FY+1,FX+1,T)
500 IF T=96 THEN 510 ELSE 52
0
510 IF BZ<2 THEN 800
520 CALL HCHAR(FY,FX,112)
530 CALL HCHAR(LY,LX,32)
540 PZ=PZ+1
550 CALL KEY(0,KEY,S)
560 IF S=0 THEN 610
570 IF BZ>0 THEN 620
580 BX=FY
590 BX=FX
600 BZ=1
610 IF BZ=0 THEN 730
620 CALL HCHAR(BY,BX,32)
630 BX=BX+1
640 BY=BY+1
650 IF BX<32 THEN 680
660 BX=1
670 BY=BY+1
680 IF BY>22 THEN 760
690 CALL GCHAR(BY,BX,X)
700 IF X=96 THEN 760
710 CALL HCHAR(BY,BX,42)
720 BZ=BZ+1
730 IF FY<23 THEN 380
740 IF FX<31 THEN 380
750 GOTO 210
760 CALL VCHAR(1,BX,32,BY)
770 CALL SOUND(-2000,-5,0)
780 BZ=0
790 GOTO 730
800 DRUCK$="PUNKTZAHL = "&ST
R$(PZ)
810 VTAB=1
820 HTAB=2
830 GOSUB 960
840 IF HS>PZ THEN 870
850 HS=PZ
860 GOSUB 1140
870 DRUCK$="HIGHSCORE= "&ST
R$(HS)
880 GOSUB 960
890 DRUCK$="NOCH EIN SPIEL?
<J> <N> "
900 GOSUB 960
910 CALL KEY(0,KEY,STATUS)
920 IF STATUS=0 THEN 910
930 IF KEY=74 THEN 190
940 IF KEY<>78 THEN 910
950 END
960 REM PRINTROUTINE
970 FOR DRI=1 TO LEN(DRUCK$)
980 CODE=ASC(SEG$(DRUCK$,DRI
,1))
990 GOSUB 1030
1000 NEXT DRI
1010 GOSUB 1080
1020 RETURN
1030 REM COUT
1040 CALL HCHAR(VTAB,HTAB,CO
DE)
1050 HTAB=HTAB+1
1060 IF HTAB<32 THEN 1130
1070 REM CROUT
1080 VTAB=VTAB+1
1090 HTAB=2
1100 IF VTAB<25 THEN 1130
1110 VTAB=24
1120 PRINT
1130 RETURN
1140 FOR SI=1 TO 5
1150 FOR SJ=-3 TO -1
1160 CALL SOUND(100,SJ,1)
1170 NEXT SJ
1180 NEXT SI
1190 RETURN
```

Dallas geht, Texas kommt

Sein Basic ist zwar äusserst langsam, aber trotzdem hat der TI-99/4A viele Anwender gefunden.
Bevor wir im Mai-Heft zum großen Texas-Festival mit 5 TI-99-Programmen einladen, in diesen Heft wieder ein einfacheres Reaktionsspiel:

Asteroiden



Zwei Tasten (Q und A) stehen den Spieler zur Verfügung, um sein UFO durch den Asteroidengürtel zu steuern.
Das Programm läuft auf allen TI-99/4A - das Extended Basic Modul ist nicht erforderlich.
Der Spielaufbau ist sehr einfach.
Bis zur Zeile 160 werden die Zeichen definiert und eingefärbt.
Ab Zeile 170 beginnt dann das eigentliche Spiel. Es werden per Zufallszahlen Asteroiden auf den Bildschirm gebracht, die das Raumschiff des Spielers umfliegen muß.
Wird es getroffen, so endet das Spiel, was ebenfalls passiert, wenn zu hoch oder zu niedrig geflogen wird.
Die Zeilen 800 bis 1190 sind mit denen im Spiel aus Heft 3/83 identisch (bis auf die Korrektur zum richigen Bildschirmausdruck). Hier wird wieder mittels der HCHAR-Funktion Text auf den Bildschirm geschrieben.

```
100 RANDOMIZE
110 CALL CLEAR
120 CALL CHAR(96,"0000000018
7E997E")
130 CALL CHAR(152,"3C7EFFFFF
F7E3C")
140 CALL CHAR(153,"1C3E7F7FF
FFF7E3C")
150 CALL COLOR(9,2,1)
160 CALL COLOR(16,9,1)
170 PZ=0
180 NUFOX=1
190 NUFOY=12
200 UFOX=NUFOX
210 UFOY=NUFOY
220 A1=RND*14+1
230 A2=RND*15+15
240 CALL GCHAR(UFOY+1,UFOX,T
)
250 IF T=152 THEN 800
260 IF T=153 THEN 800
270 PZ=PZ+1
280 CALL HCHAR(UFOY,UFOX,32,
1)
290 UFOX=NUFOX
300 UFOY=NUFOY
310 CALL HCHAR(24,A1,152,1)
320 CALL HCHAR(24,A2,153,1)
330 PRINT
340 CALL HCHAR(NUFOY,NUFOX,9
6,1)
350 CALL KEY(0,K,ST)
360 IF ST=0 THEN 390
370 IF K=81 THEN 430
380 IF K=65 THEN 460
390 NUFOX=NUFOX+1
400 IF NUFOX<33 THEN 420
410 NUFOX=1
420 GOTO 220
430 NUFOY=NUFOY-1
440 IF NUFOY<1 THEN 800
450 GOTO 390
460 NUFOY=NUFOY+1
470 IF NUFOY>23 THEN 800
480 GOTO 390
800 DRUCK$="PUNKTZAHL = "&ST
R$(PZ)
810 VTAB=1
820 HTAB=2
830 GOSUB 960
840 IF HS>PZ THEN 870
850 HS=PZ
860 GOSUB 1140
870 DRUCK$="HIGHSCORE = "&ST
R$(HS)
880 GOSUB 960
890 DRUCK$="NOCH EIN SPIEL?
<J> <N> "
900 GOSUB 960
910 CALL KEY(0,KEY,STATUS)
920 IF STATUS=0 THEN 910
930 IF KEY=74 THEN 190
940 IF KEY<>78 THEN 910
950 END
960 REM PRINTROUTINE
970 FOR DRI=1 TO LEN(DRUCK$)
980 CODE=ASC(SEG$(DRUCK$,DRI
,1))
990 GOSUB 1030
1000 NEXT DRI
1010 GOSUB 1080
1020 RETURN
1030 REM COUT
1040 CALL HCHAR(VTAB,HTAB,CO
DE)
1050 HTAB=HTAB+1
1060 IF HTAB<32 THEN 1130
1070 REM CROUT
1080 VTAB=VTAB+1
1090 HTAB=2
1100 IF VTAB<25 THEN 1130
1110 VTAB=24
1120 PRINT
1130 RETURN
1140 FOR SI=1 TO 5
1150 FOR SJ=-3 TO -1
1160 CALL SOUND(100,SJ,1)
1170 NEXT SJ
1180 NEXT SI
1190 RETURN
```

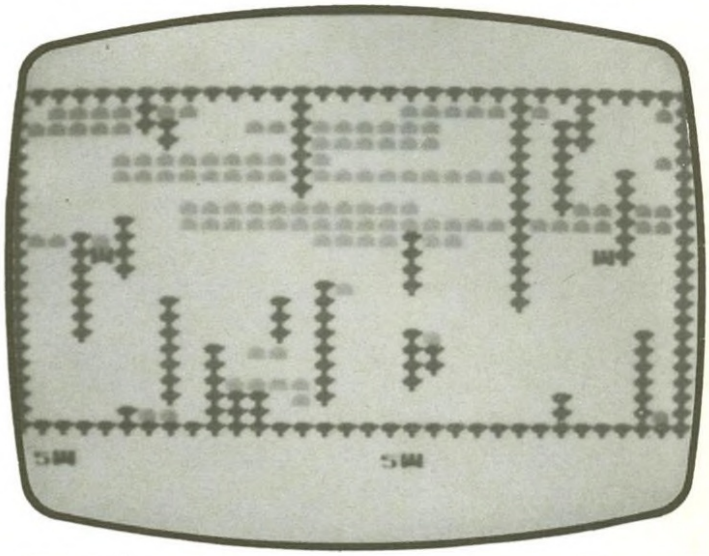
Panzerkrieg

TI-99/4A + Fernbedienungen

Der Sinn dieses Zwei-Personen-Spieles besteht darin, die gegnerischen Panzer abzuschießen und dabei möglichst wenig Flur- und Sachschaden anzurichten. Am Anfang stehen sich die Feinde gegenüber, getrennt durch Baum- und Häuserreihen. Die Häuser können zwar abgeschossen werden, bringen aber 300 Minuspunkte ein. Will man ein Haus umfahren, wird der Panzer zerstört und der Gegner bekommt 1000 Pluspunkte. Ebenso bei Abschuß. Die Bäume können weder abgeschossen noch umgefahren werden, "wehren" sich jedoch mit Minuspunkten.

Anm.: Die Zeilen 740-930 verbinden, daß ein Panzer am Anfang von Dämmen umgeben und bewegungsunfähig ist. Sollten die Gegner jedoch durch eine durchgehende Baumreihe getrennt sein, muß das Spiel mit "CLEAR" abgebrochen und mit "RUN" neu gestartet werden.

Jürgen Hoppstädter



```
>LIST -210
 100 RANDOMIZE
 110 REM  ZEICHEN DEFINIEREN
 120 CALL CHAR(128,"DBDBDBFFF
FFFFFC3")
 130 CALL CHAR(129,"FFFF787F7
F78FFFF")
 140 CALL CHAR(130,"C3FFFFFFF
FDBDBDB")
 150 CALL CHAR(131,"FFFF1EFEF
E1EFFFF")
 160 CALL CHAR(132,"000000181
8000000")
 170 CALL CHAR(133,"00003C242
43C")
 180 CALL CHAR(134,"007E42424
2427E")
 190 CALL CHAR(135,"FF8181818
18181FF")
 200 CALL CHAR(136,"DBDBDBFFF
FFFFFC3")
 210 CALL CHAR(137,"FFFF787F7
F78FFFF")
>
```

```
 220 CALL CHAR(138,"C3FFFFFFF
FDBDBDB")
 230 CALL CHAR(139,"FFFF1EFEF
E1EFFFF")
 240 CALL CHAR(140,"000000181
8000000")
 250 CALL CHAR(141,"00003C242
43C")
 260 CALL CHAR(142,"007E42424
2427E")
 270 CALL CHAR(143,"FF8181818
18181FF")
 280 CALL CHAR(144,"3C7FFF7E7
C181818")
 290 CALL CHAR(152,"00387CFEF
EFEFEFE")
 300 CALL CHAR(153,"00003C242
43C")
 310 CALL CHAR(154,"007E42424
2427E")
 320 CALL CHAR(155,"FF8181818
18181FF")
 330 REM  FARBEN DEFINIEREN
>
```

```
>LIST 340-550
 340 CALL SCREEN(15)
 350 CALL COLOR(13,5,1)
 360 CALL COLOR(14,14,1)
 370 CALL COLOR(15,13,1)
 380 CALL COLOR(16,10,1)
 390 REM  HAEUSER BAUEN
 400 CALL CLEAR
 410 P1=0
 420 P2=0
 430 FOR I=1 TO 30
 440 X=23*RND+1
 450 Y=31*RND+1
 460 Z=9*RND+1
 470 CALL HCHAR(X,Y,152,Z)
 480 NEXT I
 490 REM  BAEUME PFLANZEN
 500 FOR I=1 TO 30
```

```
 510 X=23*RND+1
 520 Y=31*RND+1
 530 Z=7*RND+1
 540 CALL VCHAR(X,Y,144,Z)
 550 NEXT I
>
```

```
>LIST 560-750
 560 REM  RAND/PUNKTETABELLE
 570 CALL HCHAR(24,1,32,32)
 580 CALL HCHAR(23,1,32,32)
 590 PANZER1=5
 600 PANZER2=5
 610 GOSUB 4560
 620 GOSUB 4590
 630 CALL HCHAR(1,1,144,32)
 640 CALL HCHAR(22,1,144,32)
 650 CALL VCHAR(1,1,144,22)
 660 CALL VCHAR(1,32,144,22)
 670 CALL HCHAR(24,3,128)
 680 CALL HCHAR(24,19,136)
 690 REM  STARTPOSITION
 700 XP1=11
 710 XP2=11
 720 YP1=5
 730 YP2=28
 740 CALL GCHAR(XP1-1,YP1,ZP1
)
 750 IF ZP1<>144 THEN 840
>
```

```
 760 CALL GCHAR(XP1,YP1+1,ZP1
)
 770 IF ZP1<>144 THEN 840
 780 CALL GCHAR(XP1+1,YP1,ZP1
)
 790 IF ZP1<>144 THEN 840
 800 CALL GCHAR(XP1,YP1-1,ZP1
)
 810 IF ZP1<>144 THEN 840
 820 YP1=YP1+1
 830 GOTO 740
 840 CALL GCHAR(XP2-1,YP2,ZP2
)
 850 IF ZP2<>144 THEN 940
 860 CALL GCHAR(XP2,YP2+1,ZP2
)
 870 IF ZP2<>144 THEN 940
 880 CALL GCHAR(XP2+1,YP2,ZP2
)
 890 IF ZP2<>144 THEN 940
 900 CALL GCHAR(XP2,YP2-1,ZP2
)
 910 IF ZP2<>144 THEN 940
>
```

```
 920 YP2=YP2-1
 930 GOTO 840
 940 CALL HCHAR(XP1,YP1,128)
 950 CALL HCHAR(XP2,YP2,136)
 960 REM  SPIELHAUPTPROGRAM
M
 970 CALL GCHAR(XP1,YP1,ZP1)
 980 CALL JOYST(1,X1,Y1)
 990 IF Y1<>4 THEN 1010
1000 GOSUB 1240
```

```
1010  IF X1<>4 THEN 1030
1020  GOSUB 1380
1030  IF Y1<>-4 THEN 1050
1040  GOSUB 1510
1050  IF X1<>-4 THEN 1070
1060  GOSUB 1640
1070  CALL KEY(2,K2,S2)
1080  IF K2<>18 THEN 1100
1090  GOSUB 3540
1100  CALL KEY(1,K1,S1)
1110  IF K1<>18 THEN 1130
1120  GOSUB 2660
1130  CALL GCHAR(XP2,YP2,ZP2)
>■
```

```
1140  CALL JOYST(2,X2,Y2)
1150  IF Y2<>4 THEN 1170
1160  GOSUB 1770
1170  IF X2<>4 THEN 1190
1180  GOSUB 1910
1190  IF Y2<>-4 THEN 1210
1200  GOSUB 2040
1210  IF X2<>-4 THEN 1230
1220  GOSUB 2170
1230  GOTO 970
1240  REM  FORTBEW. PANZER 1

1250  REM  NACH OBEN
1260  IF ZP1=128 THEN 1300
1270  CALL SOUND(-1000,-8,5)
1280  CALL HCHAR(XP1,YP1,128)
1290  GOTO 1370
1300  CALL GCHAR(XP1-1,YP1,H1
)
1310  IF H1=144 THEN 2300
1320  CALL HCHAR(XP1,YP1,32)
1330  IF H1=152 THEN 2340
1340  XP1=XP1-1
>
```

```
>LIST 1350-1550
1350  CALL SOUND(-1000,-8,5)
1360  CALL HCHAR(XP1,YP1,128)
1370  RETURN
1380  REM  NACH RECHTS
1390  IF ZP1=129 THEN 1430
1400  CALL SOUND(-1000,-8,5)
1410  CALL HCHAR(XP1,YP1,129)
1420  GOTO 1500
1430  CALL GCHAR(XP1,YP1+1,H1
)
1440  IF H1=144 THEN 2300
1450  CALL HCHAR(XP1,YP1,32)
1460  IF H1=152 THEN 2340
1470  YP1=YP1+1
1480  CALL SOUND(-1000,-8,5)
1490  CALL HCHAR(XP1,YP1,129)
1500  RETURN
1510  REM  NACH UNTEN
1520  IF ZP1=130 THEN 1560
1530  CALL SOUND(-1000,-8,5)
1540  CALL HCHAR(XP1,YP1,130)
1550  GOTO 1630
>
```

```
1560  CALL GCHAR(XP1+1,YP1,H1
)
1570  IF H1=144 THEN 2300
1580  CALL HCHAR(XP1,YP1,32)
1590  IF H1=152 THEN 2340
1600  XP1=XP1+1
1610  CALL SOUND(-1000,-8,5)
1620  CALL HCHAR(XP1,YP1,130)
1630  RETURN
1640  REM  NACH UNTEN
1650  IF ZP1=131 THEN 1690
1660  CALL SOUND(-1000,-8,5)
1670  CALL HCHAR(XP1,YP1,131)
1680  GOTO 1760
1690  CALL GCHAR(XP1,YP1-1,H1
)
1700  IF H1=144 THEN 2300
1710  CALL HCHAR(XP1,YP1,32)
1720  IF H1=152 THEN 2340
1730  YP1=YP1-1
1740  CALL SOUND(-1000,-8,5)
1750  CALL HCHAR(XP1,YP1,131)
1760  RETURN
>■
```

```
1770  REM  FORTBEW. PANZER 2
1780  REM  NACH OBEN
1790  IF ZP2=136 THEN 1830
1800  CALL SOUND(-1000,-4,5)
1810  CALL HCHAR(XP2,YP2,136)
1820  GOTO 1900
1830  CALL GCHAR(XP2-1,YP2,H2
)
1840  IF H2=144 THEN 2480
1850  CALL HCHAR(XP2,YP2,32)
1860  IF H2=152 THEN 2520
1870  XP2=XP2-1
1880  CALL SOUND(-1000,-4,5)
1890  CALL HCHAR(XP2,YP2,136)
```

```
1900  RETURN
1910  REM  NACH RECHTS
1920  IF ZP2=137 THEN 1960
1930  CALL SOUND(-1000,-4,5)
1940  CALL HCHAR(XP2,YP2,137)
1950  GOTO 2030
1960  CALL GCHAR(XP2,YP2+1,H2
)
1970  IF H2=144 THEN 2480
>■
```

```
1980  CALL HCHAR(XP2,YP2,32)
1990  IF H2=152 THEN 2520
2000  YP2=YP2+1
2010  CALL SOUND(-1000,-4,5)
2020  CALL HCHAR(XP2,YP2,137)
2030  RETURN
2040  REM  NACH UNTEN
2050  IF ZP2=138 THEN 2090
2060  CALL SOUND(-1000,-4,5)
2070  CALL HCHAR(XP2,YP2,138)
2080  GOTO 2160
2090  CALL GCHAR(XP2+1,YP2,H2
)
2100  IF H2=144 THEN 2480
2110  CALL HCHAR(XP2,YP2,32)
2120  IF H2=152 THEN 2520
2130  XP2=XP2+1
2140  CALL SOUND(-1000,-4,5)
2150  CALL HCHAR(XP2,YP2,138)
2160  RETURN
2170  REM  NACH LINKS
2180  IF ZP2=139 THEN 2220
2190  CALL SOUND(-1000,-4,5)
>
```

```
2200  CALL HCHAR(XP2,YP2,139)
2210  GOTO 2290
2220  CALL GCHAR(XP2,YP2-1,H2
)
2230  IF H2=144 THEN 2480
2240  CALL HCHAR(XP2,YP2,32)
2250  IF H2=152 THEN 2520
2260  YP2=YP2-1
2270  CALL SOUND(-1000,-4,5)
2280  CALL HCHAR(XP2,YP2,139)
2290  RETURN
2300  CALL SOUND(500,110,5)
2310  P1=P1-100
2320  GOSUB 4420
2330  RETURN
2340  CALL SOUND(400,-7,5)
2350  CALL HCHAR(XP1,YP1,133)
2360  CALL SOUND(600,-6,5)
2370  CALL HCHAR(XP1,YP1,134)
2380  CALL SOUND(800,-5,5)
2390  CALL HCHAR(XP1,YP1,135)
2400  CALL HCHAR(XP1,YP1,32)
2410  CALL HCHAR(XP2,YP2,32)
>■
```

```
2420  P2=P2+1000
2430  PANZER1=PANZER1-1
2440  GOSUB 4490
2450  GOSUB 4560
2460  IF PANZER1<1 THEN 4620
2470  GOTO 700
2480  CALL SOUND(500,110,5)
2490  P2=P2-100
2500  GOSUB 4490
2510  RETURN
2520  CALL SOUND(400,-7,5)
2530  CALL HCHAR(XP2,YP2,141)
2540  CALL SOUND(600,-6,5)
2550  CALL HCHAR(XP2,YP2,142)
2560  CALL SOUND(800,-5,5)
2570  CALL HCHAR(XP2,YP2,143)
2580  CALL HCHAR(XP2,YP2,32)
2590  CALL HCHAR(XP1,YP1,32)
2600  P1=P1+1000
2610  PANZER2=PANZER2-1
2620  GOSUB 4420
2630  GOSUB 4590
2640  IF PANZER2<1 THEN 4620
>■
```

```
2650  GOTO 700
2660  REM  PANZER 1 SCHUSS
2670  CALL GCHAR(XP1,YP1,ZP1)
2680  XK1=XP1
2690  YK1=YP1
2700  ON ZP1-127 GOSUB 2720,3
000,3180,3360
2710  RETURN
2720  REM  NACH OBEN
2730  XK1=XK1-1
2740  CALL GCHAR(XK1,YK1,ZK1)
2750  IF ZK1<>32 THEN 2800
2760  CALL HCHAR(XK1,YK1,132)
2770  CALL SOUND(100,-4,5)
2780  CALL HCHAR(XK1,YK1,32)
2790  GOTO 2720
2800  IF ZK1<>144 THEN 2850
```

```
2810 CALL SOUND(100,110,5)
2820 P1=P1-50
2830 GOSUB 4420
2840 GOTO 2710
2850 IF ZK1<>152 THEN 2880
2860 GOSUB 2900
>
```

```
2870 GOTO 2710
2880 GOSUB 2520
2890 GOTO 2710
2900 CALL SOUND(600,-5,5)
2910 CALL HCHAR(XK1,YK1,153)
2920 CALL SOUND(700,-6,5)
2930 CALL HCHAR(XK1,YK1,154)
2940 CALL SOUND(900,-7,5)
2950 CALL HCHAR(XK1,YK1,155)
2960 P1=P1-300
2970 GOSUB 4420
2980 CALL HCHAR(XK1,YK1,32)
2990 RETURN
3000 REM NACH RECHTS
3010 YK1=YK1+1
3020 CALL GCHAR(XK1,YK1,ZK1)
3030 IF ZK1<>32 THEN 3080
3040 CALL HCHAR(XK1,YK1,132)
3050 CALL SOUND(100,-4,5)
3060 CALL HCHAR(XK1,YK1,32)
3070 GOTO 3000
3080 IF ZK1<>144 THEN 3130
3090 CALL SOUND(100,110,5)
>
```

```
3100 P1=P1-50
3110 GOSUB 4420
3120 GOTO 2710
3130 IF ZK1<>152 THEN 3160
3140 GOSUB 2900
3150 GOTO 2710
3160 GOSUB 2520
3170 GOTO 2710
3180 REM NACH UNTEN
3190 XK1=XK1+1
3200 CALL GCHAR(XK1,YK1,ZK1)
3210 IF ZK1<>32 THEN 3260
3220 CALL HCHAR(XK1,YK1,132)
3230 CALL SOUND(100,-4,5)
3240 CALL HCHAR(XK1,YK1,32)
3250 GOTO 3180
3260 IF ZK1<>144 THEN 3310
3270 CALL SOUND(100,110,5)
3280 P1=P1-50
3290 GOSUB 4420
3300 GOTO 2710
3310 IF ZK1<>152 THEN 3340
3320 GOSUB 2900
>
```

```
3330 GOTO 2710
3340 GOSUB 2520
3350 GOTO 2710
3360 REM NACH LINKS
3370 YK1=YK1-1
3380 CALL GCHAR(XK1,YK1,ZK1)
3390 IF ZK1<>32 THEN 3440
3400 CALL HCHAR(XK1,YK1,132)
3410 CALL SOUND(100,-4,5)
3420 CALL HCHAR(XK1,YK1,32)
3430 GOTO 3360
3440 IF ZK1<>144 THEN 3310
3450 CALL SOUND(100,110,5)
3460 P1=P1-50
3470 GOSUB 4420
3480 GOTO 2710
3490 IF ZK1<>152 THEN 3520
3500 GOSUB 2900
3510 GOTO 2710
3520 GOSUB 2520
3530 GOTO 2710
3540 REM PANZER 2 SCHUSS
3550 CALL GCHAR(XP2,YP2,ZP2)
>
```

```
3560 XK2=XP2
3570 YK2=YP2
3580 ON ZP2-135 GOSUB 3600,3
780,3960,4140
3590 RETURN
3600 REM NACH OBEN
3610 XK2=XK2-1
3620 CALL GCHAR(XK2,YK2,ZK2)
3630 IF ZK2<>32 THEN 3680
3640 CALL HCHAR(XK2,YK2,140)
3650 CALL SOUND(100,-4,5)
3660 CALL HCHAR(XK2,YK2,32)
3670 GOTO 3600
3680 IF ZK2<>144 THEN 3730
3690 CALL SOUND(100,110,5)
3700 P2=P2-50
3710 GOSUB 4490
3720 GOTO 3590
3730 IF ZK2<>152 THEN 3760
3740 GOSUB 4320
3750 GOTO 3590
```

```
3760 GOSUB 2340
3770 GOTO 3590
>
```

```
3780 REM NACH RECHTS
3790 YK2=YK2+1
3800 CALL GCHAR(XK2,YK2,ZK2)
3810 IF ZK2<>32 THEN 3860
3820 CALL HCHAR(XK2,YK2,140)
3830 CALL SOUND(100,-4,5)
3840 CALL HCHAR(XK2,YK2,32)
3850 GOTO 3780
3860 IF ZK2<>144 THEN 3910
3870 CALL SOUND(100,110,5)
3880 P2=P2-50
3890 GOSUB 4490
3900 GOTO 3590
3910 IF ZK2<>152 THEN 3940
3920 GOSUB 4320
3930 GOTO 3590
3940 GOSUB 2340
3950 GOTO 3590
3960 REM NACH UNTEN
3970 XK2=XK2+1
3980 CALL GCHAR(XK2,YK2,ZK2)
3990 IF ZK2<>32 THEN 4040
4000 CALL HCHAR(XK2,YK2,140)
>
```

```
4010 CALL SOUND(100,-4,5)
4020 CALL HCHAR(XK2,YK2,32)
4030 GOTO 3960
4040 IF ZK2<>144 THEN 4090
4050 CALL SOUND(100,110,5)
4060 P2=P2-50
4070 GOSUB 4490
4080 GOTO 3590
4090 IF ZK2<>152 THEN 4120
4100 GOSUB 4320
4110 GOTO 3590
4120 GOSUB 2340
4130 GOTO 3590
4140 REM NACH LINKS
4150 YK2=YK2-1
4160 CALL GCHAR(XK2,YK2,ZK2)
4170 IF ZK2<>32 THEN 4220
4180 CALL HCHAR(XK2,YK2,140)
4190 CALL SOUND(100,-4,5)
4200 CALL HCHAR(XK2,YK2,32)
4210 GOTO 4140
4220 IF ZK2<>144 THEN 4270
4230 CALL SOUND(100,110,5)
>
```

```
4240 P2=P2-50
4250 GOSUB 4490
4260 GOTO 3590
4270 IF ZK2<>152 THEN 4300
4280 GOSUB 4320
4290 GOTO 3590
4300 GOSUB 2340
4310 GOTO 3590
4320 CALL SOUND(600,-5,5)
4330 CALL HCHAR(XK2,YK2,153)
4340 CALL SOUND(700,-6,5)
4350 CALL HCHAR(XK2,YK2,154)
4360 CALL SOUND(900,-7,5)
4370 CALL HCHAR(XK2,YK2,155)
4380 P2=P2-300
4390 GOSUB 4490
4400 CALL HCHAR(XK2,YK2,32)
4410 RETURN
4420 REM PUNKTE PANZER 1
4430 D1$=STR$(P1)&" P "
4440 FOR I=1 TO LEN(D1$)
4450 Z1=ASC(SEG$(D1$,I,1))
4460 CALL HCHAR(24,4+I,Z1)
>
```

```
>LIST 4470-4650
4470 NEXT I
4480 RETURN
4490 REM PUNKTE PANZER 2
4500 D2$=STR$(P2)&" P "
4510 FOR I=1 TO LEN(D2$)
4520 Z2=ASC(SEG$(D2$,I,1))
4530 CALL HCHAR(24,20+I,Z2)
4540 NEXT I
4550 RETURN
4560 PANZER1$=STR$(PANZER1)
4570 CALL HCHAR(24,2,ASC(PAN
ZER1$))
4580 RETURN
4590 PANZER2$=STR$(PANZER2)
4600 CALL HCHAR(24,18,ASC(PA
NZER2$))
4610 RETURN
4620 PRINT
4630 PRINT
4640 PRINT "SPIELENDE"
4650 PRINT
>
```

Fortsetzung auf Seite 61

TI-UFO

Ziel des Spieles ist es, ein Ufo um Hindernisse zu lenken. Die horizontale Steuerung erfolgt mit den Tasten A (nach links) und L (nach rechts). Wird keine Taste gedrückt, so bewegt sich das Ufo vertikal nach unten. Wird ein Hindernis senkrecht überfahren, so werden 2 Fehlerpunkte berechnet, beim waagrechten Überfahren 1 Fehlerpunkt. Ebenso werden beim waagrechten Zurückfahren und bei Erreichen des Bildschirmrandes Fehlerpunkte errechnet.

Die ersten drei Runden bestehen aus drei Durchgängen. Jede weitere Runde aus zwei Durchgängen. Der Schwierigkeitsgrad nimmt ständig zu. Werden in einer Runde weniger als 8 Fehlerpunkte verursacht, so kommt man in die nächste Runde (Freispiel). Der Schwierigkeitsgrad kann in Zeile 170, der Schwierigkeitszuwachs in der Zeile 770 verstellt werden.

Runden 1-3: # $ %
Runden 4 - 8: A B C D E
Runden 9 - : Zufallszeichen

Nachteil des Programms: Das Ufo kann aus Zeitgründen nicht gelöscht werden. Es bildet sich deshalb eine Ufoschlange.

Hans-Peter Adrion

```
100 CALL SCREEN(14)
110 CALL COLOR(1,16,14)
120 BU=1
130 JU=0
140 FE=0
150 ZE=35
160 REM HINDERNISSE PRO ZEIL
E
170 PIT=5
180 SC=0
190 FOR T=3 TO BU STEP -1
200 IF FE>=8 THEN 1050
210 CALL CLEAR
220 G=0
230 RANDOMIZE
240 GOSUB 560
250 IF FE>=8 THEN 1050
260 CALL SOUND(400,131,2,165
,2,196,2)
270 REM  **UFO SETZEN
280 SPAL=16
290 CALL CHAR(130,"000000183
C7E7E99")
300 G=G+1
310 CALL GCHAR(G,SPAL,WE)
320 IF WE>33 THEN 330 ELSE 3
60
330 CALL SOUND(100,-2,1)
340 CALL COLOR(1,5,14)
350 SC=SC+2
360 CALL HCHAR(G,SPAL,130)
370 IF G=24 THEN 660
380 CALL KEY(0,KEY,STATUS)
390 IF STATUS=0 THEN 300
400 IF KEY=65 THEN 460
410 IF KEY=76 THEN 430
420 GOTO 300
430 SPAL=SPAL+1
440 IF SPAL=33 THEN 460
450 GOTO 480
460 SPAL=SPAL-1
470 IF SPAL=0 THEN 430
480 CALL GCHAR(G,SPAL,Q)
490 IF Q>33 THEN 500 ELSE 53
0
500 CALL SOUND(100,-1,1)
510 CALL COLOR(1,16,14)
520 SC=SC+1
530 CALL HCHAR(G,SPAL,130)
540 GOTO 380
550 REM **HINDERNISSE SETZEN
560 Z=3
570 FOR M=1 TO 100
580 FOR C=1 TO PIT
590 X=INT(32*RND)+1
600 IF Z>24 THEN 650
610 CALL HCHAR(Z,X,ZE)
620 NEXT C
630 Z=Z+T
640 NEXT M
650 RETURN
660 CALL SOUND(2500,131,3,17
5,3,220,3)
670 NEXT T
680 CALL CLEAR
690 FE=FE+1
700 PRINT
710 REM **ASCII-CODE
720 PRINT
730 ZE=ZE+1
740 PRINT
750 PRINT
760 REM HINDERNISZUWACHS
770 PIT=PIT+4
780 REM AUSGABE
790 PRINT "FEHLERPUNKTE :";S
C
800 JU=JU+1
810 IF JU=3 THEN 1020
820 REM FREISPIEL SETZEN!
830 IF SC<=8 THEN 840 ELSE 9
30
840 PRINT
850 IF SC=0 THEN 860 ELSE 87
0
860 PRINT " O P T I M U M !"
870 PRINT
880 PRINT " ***FREISPIEL***"
890 REM PAUSE
900 FOR V=1 TO 800
910 NEXT V
920 GOTO 180
930 PRINT
940 PRINT "C BY HOMECOMPUTER
"
950 PRINT
960 PRINT "NOCH EIN SPIEL? (
J/N)"
970 CALL KEY(0,KEY,STATUS)
980 IF STATUS=0 THEN 970
990 IF KEY=74 THEN 120
1000 STOP
1010 REM ZEICHENSETZUNG AB
        RUNDE 8
1020 BU=2
1030 ZE=65
1040 GOTO 830
1050 CALL CLEAR
1060 TR=TR+50
1070 FOR OP=1 TO TR
1080 WA=INT(120*RND)+1
1090 WS=INT(22:RND)+3
1100 WD=INT(32*RND)+1
1110 CALL HCHAR(WS,WD,WA)
1120 NEXT OP
1130 G=0
1140 GOTO 260
1150 END
```

Killersatellit

Sobald das Spiel gestartet wurde, fragt der Computer nach der Wahl des Schiffes. Es stehen 4 Schiffe zur Wahl, die sich jedoch nur äußerlich unterscheiden. Die Steuerung des Schiffes erfolgt über die 4 Cursor-Tasten (E, S, D und X). Über die Space-Taste können Schüsse abgegeben werden. Nach jedem erfolgtem Abschuß erhöht sich die Schwierigkeit und Punktzahl der einzelnen Gegner. Außerdem wird nach jedem Abschuß die eigene, sowie die Höchstpunktzahl angezeigt. Das Spiel ist beendet, wenn der Satellit das Schiff abgeschossen hat. Alles Andere ergibt sich beim Spielen.

Ch. Bladoschewski

```
10 REM ********************
20 REM *  KILLERSATELLIT   *
30 REM *  ENTWICKELT VON   *
40 REM * CH. BLADOSCHEWSKI *
50 REM ********************
100 CALL CLEAR
110 CALL COLOR(1,2,2)
120 CALL SCREEN(2)
130 PRINT TAB(7);"KILLERSATE
LLIT": : : : : : : : : : : :
140 CALL CHAR(96,"00183C7E7E
E3C18")
150 CALL CHAR(104,"00183C7E7
E3C18")
160 CALL HCHAR(10,6,104,21)
170 CALL HCHAR(14,6,104,21)
180 CALL VCHAR(10,6,104,5)
190 CALL VCHAR(10,26,104,5)
200 FOR I=6 TO 26 STEP 2
210 CALL HCHAR(10,I,96)
220 CALL HCHAR(14,I,96)
230 NEXT I
240 CALL HCHAR(12,6,96)
250 CALL HCHAR(12,26,96)
260 CALL COLOR(1,16,1)
270 CALL SCREEN(16)
280 FOR I=1 TO 20
290 FOR J=1 TO 2
300 CALL COLOR(9,11,9)
310 CALL COLOR(10,5,13)
320 NEXT J
330 CALL COLOR(9,5,9)
340 CALL COLOR(10,11,13)
350 NEXT I
360 CALL COLOR(1,1,3)
370 CALL SCREEN(3)
380 REM
390 SC=0
400 CALL CLEAR
410 RANDOMIZE
420 M=2
430 P1=1
440 U=.1
450 A$="1010103BABAFE82"
460 R$="42425A665A4242"
470 C$="183C7E3C3C7EFF"
480 D$="82828200828282"
490 E$="1818181818181818"
500 B$="1818183C66FFFF66"
510 G$="42425A5A7EFFFF42"
520 INPUT "WELCHES SCHIFF WI
LLST DU (1-4)?":SCHIFF
530 IF (SCHIFF<1)+(SCHIFF>4)
<0 THEN 520
540 ON SCHIFF GOTO 640,580,6
10,550
550 Z$=R$
560 Y$=E$
570 GOTO 670
580 Z$=B$
590 Y$=E$
600 GOTO 670
610 Y$=D$
620 Z$=G$
630 GOTO 660
640 REM
650 Z$=A$
660 Y$=D$
670 X=10
680 Y=10
690 CALL CLEAR
700 B=12
710 A=16
720 CALL SCREEN(2)
730 CALL COLOR(1,1,1)
740 CALL COLOR(9,4,1)
750 CALL COLOR(10,14,1)
760 CALL COLOR(11,9,1)
770 CALL COLOR(12,5,1)
780 CALL CHAR(96,Y$)
790 CALL CHAR(104,"5A5A5A5A5
A5A5A5A")
800 CALL CHAR(112,C$)
810 CALL CHAR(120,Z$)
820 X=X-(INT((RND*2)-.5))
830 Y=Y-(INT((RND*2)-.5))
840 IF X<1 THEN 1240
850 IF X>24 THEN 1260
860 IF Y<1 THEN 1280
870 IF Y>32 THEN 1300
880 CALL KEY(0,K,S)
890 IF K=32 THEN 1320
900 IF K=83 THEN 1040
910 IF K=68 THEN 1070
920 IF K=88 THEN 1100
930 IF K=69 THEN 1130 ELSE 9
80
940 IF B<1 THEN 1160
950 IF B>24 THEN 1180
960 IF A<1 THEN 1200
970 IF A>32 THEN 1220
980 CALL CLEAR
990 CALL HCHAR(B,A,120)
1000 CALL HCHAR(X,Y,112)
1010 IF X>21 THEN 820
1020 IF RND<U THEN 1580
1030 GOTO 820
1040 A=A-1
1050 CALL SOUND(-500,-6,0)
1060 GOTO 940
1070 A=A+1
1080 CALL SOUND(-500,-6,0)
1090 GOTO 940
1100 B=B+1
1110 CALL SOUND(-500,-6,0)
1120 GOTO 940
1130 B=B-1
1140 CALL SOUND(-500,-6,0)
1150 GOTO 940
1160 B=B+24
1170 GOTO 980
1180 B=B-24
1190 GOTO 980
1200 A=A+32
1210 GOTO 980
1220 A=A-32
1230 GOTO 980
1240 X=X+24
1250 GOTO 880
1260 X=X-24
1270 GOTO 880
1280 Y=Y+32
1290 GOTO 880
1300 Y=Y-32
1310 GOTO880
1320 FOR P=B-1 TO B-M STEP -
1
1330 CALL SOUND(100,-5,0)
1340 CALL GCHAR(P,A,L)
1350 IF L=112 THEN 1390
1360 CALL HCHAR(P,A,96)
1370 NEXT P
1380 GOTO 980
1390 CALL CLEAR
1400 PRINT "GEGNER ZERSTOERT
": : : : :
1410 SC=SC+P1
1420 IF HS>=SC THEN 1490
1430 HS=SC
1440 FOR SI=1 TO 5
1450 FOR SJ=-3 TO -1
1460 CALL SOUND(100,SJ,0)
1470 NEXT SJ
1480 NEXT SI
1490 PRINT "HIGHSCORE:";HS
1500 PRINT "PUNKTZAHL:";SC:
: : : : : : : : : : : :
1510 CALL COLOR(1,3,3)
1520 CALL SCREEN(3)
1530 P1=P1+1
1540 U=U+.02
1550 FOR I=1 TO 1000
1560 NEXT I
1570 GOTO 540
1580 FOR P=X+1 TO X+3
1590 CALL SOUND(100,-3,2)
1600 CALL GCHAR(P,Y,L)
1610 IF L=120 THEN 1650
1620 CALL HCHAR(P,Y,104)
1630 NEXT P
1640 GOTO 820
1650 CALL CLEAR
1660 PRINT TAB(5);"DU WURDES
T ZERSTOERT": : :"
GAME OVER": : :
1670 PRINT "HIGHSCORE:";HS
1680 PRINT "PUNKTZAHL:";SC:
: : : : : : : :
1690 CALL COLOR(1,3,3)
1700 CALL SCREEN(3)
1710 PRINT "WILLST DU NOCH E
INMAL SPIE- LEN? DRUECKE <J>
 <N>"
1720 CALL KEY(0,K,S)
1730 IF S=0 THEN 1720
1740 IF K=ASC("J")THEN 380
1750 IF K<>ASC("N")THEN 1710
1760 END
```

Chiffrier/ Dechiffrier-Programm

TI 99/4A + Drucker

Eingabe
Das Pgm fragt zunächst, ob dechiffriert werden soll, – dann nur DE eingeben (wird im Pgm auch gesagt) Ansonsten verlangt es die Eingabe einer vierstelligen Codezahl (z.B. Datum in der Form 3103 = 31.03.) Diese Codezahl muß auch dem Entschlüssler bekannt sein.
Das Pgm nimmt nur Zahlen an und zwar nur eine! Es sind also alle vier einzeln einzugeben. Danach wird die Eingabe einer zu verschlüsselnden Zeile verlangt. Das Pgm nimmt nur 28 Zeichen an und sagt an, daß der Zeilenrest ggflls mit Punkten zu füllen ist. Die Eingabe geschieht ganz normal und der Drucker druckt sie verschlüsselt aus. So kann Zeile für Zeile eingegeben werden. Die letzte Eingabe soll eine Ø sein, damit das Pgm erkennt, daß die Eingabe abgeschlossen ist.
Das Dechiffrieren geschieht genauso, nur daß hier eben der chiffrierte Text eingegeben wird, der dann Zeile für Zeile dechiffriert herauskommt.
Interessant und daher kaum zu „knacken" ist der Algorhitmus des Pgms hinsichtlich der Verschlüsselung. Manchmal ist der gleiche Buchstabe von gleicher Bedeutung, manchmal haben 2 gleiche Buchstaben eine völlig andere Bedeutung und manchmal werden für einen Buchstaben im laufe eines längeren Textes bis zu 50 andere eingesetzt.

Programmbeispiel:
Eingegeben wurde zeilenweise

DIES IST EIN CHIFFRIERTEST::
AUFGENOMMEN AM 31.03.83 FÜR DIE ZEITSCHRIFT HOMECOMPUTER

Da die Eingabe automatisch auf 28 Zeichen beschränkt ist, können Sie am Zeilenende jeweils auf dem . stehenbleiben, der Computer meldet sich mit einem BEEP.
So sah der Ausdruck aus:

```
AHEP HSQ DIK BHFCERFBQTBPS.+
>TFDBMOJJDN >L 0.-00+73 CTEO
AHE WDIQPBHOFET ENMB@NMMRSEO
```

und als der vorige Ausdruck Zeichen für Zeichen, wiederum zeilenweise nach vorheriger Eingabe der Codeziffern eingegeben wurde, kam folgender Ausdruck zustande:

```
DIES IST EIN CHIFFRIERTEST..
AUFGENOMMEN AM 31.03.83 FUER
DIE ZEITSCHRIFT HOMECOMPUTER
```

```
100 CALL CLEAR :: CALL SCREEN(16
)
110 DIM B$(30)
120 DIM B(30)
130 INPUT "ZUM DECHIFFRIEREN DE
EINGEBEN":DE$ :: IF DE$="DE" THE
N 370
140 PRINT "GIB JETZT EINZELN DIE
 4 CODEZIFFERN EIN:"
150 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):W
160 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):X
170 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Y
180 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Z
190 PRINT "TEXTZEILE EINGEBEN  (
BIS 28.STELLE MIT PUNKTEN FUELLE
N):"
200 ACCEPT BEEP SIZE(28):A$
210 IF A$="0" THEN END
220 FOR I=1 TO 28
230 B(I)=ASC(SEG$(A$,I,1))
240 NEXT I
250 FOR I=1 TO 28 STEP 4 :: B(I)
=B(I)-W :: NEXT I
260 FOR I=2 TO 28 STEP 4 :: B(I)
=B(I)-X :: NEXT I
270 FOR I=3 TO 28 STEP 4 :: B(I)
=B(I)-Y :: NEXT I
280 FOR I=4 TO 28 STEP 4 :: B(I)
=B(I)-Z :: NEXT I
290 FOR I=1 TO 28
300 IF B(I)<=32 THEN B(I)=32
310 NEXT I
320 FOR I=1 TO 28 :: B$(I)=CHR$(
B(I)):: NEXT I
330 OPEN #1:"TP.E",OUTPUT
340 FOR I=1 TO 28 :: PRINT #1:B$
(I);:: NEXT I
350 CLOSE #1
360 GOTO 200
370 PRINT "GIB JETZT EINZELN DIE
 4 CODEZIFFERN EIN:"
380 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):W
390 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):X
400 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Y
410 ACCEPT BEEP VALIDATE(DIGIT)S
IZE(1):Z
420 PRINT "GIB DEN ZEILENTEXT EI
N":A$
430 ACCEPT BEEP SIZE(28):A$
440 IF A$="0" THEN END
450 FOR I=1 TO 28
460 B(I)=ASC(SEG$(A$,I,1))
470 NEXT I
480 FOR I=1 TO 28 STEP 4 :: B(I)
=B(I)+W :: NEXT I
490 FOR I=2 TO 28 STEP 4 :: B(I)
=B(I)+X :: NEXT I
```

```
500 FOR I=3 TO 28 STEP 4 :: B(I)
=B(I)+Y :: NEXT I
510 FOR I=4 TO 28 STEP 4 :: B(I)
=B(I)+Z :: NEXT I
520 FOR I=1 TO 28
530 IF B(I)<=41 THEN B(I)=32
540 NEXT I
550 FOR I=1 TO 28 :: B$(I)=CHR$(
B(I)):: NEXT I
560 OPEN #1:"TP.E",OUTPUT
570 FOR I=1 TO 28 :: PRINT #1:B$
(I);:: NEXT I
580 CLOSE #1
590 GOTO 430
600 END
```

Rangliste

TI 99/4A + Drucker

Aufgrund einer Wahl oder eines Wettbewerbes soll im Verein z. B. eine Rangliste der Ergebnisse aufgestellt werden.

Die Verarbeitungszeit für 100 Teilnehmer beträgt ca. 5 min.

Das Programm fragt in einer Dreiergruppe für jeden Teilnehmer:
LISTENPLATZ
NAME
ERGEBNIS (in Punkten oder Stimmen)
Sind die Angaben für alle Teilnehmer eingegeben, erwartet das Pgm zum Abschluß unter LISTENPLATZ die Eingabe einer 0 und beginnt die Verarbeitung.

Ausgabeform:
1. (5.) Mayerhofer = 189
dabei ist die erste Zahl der Ranglistenplatz die in ( ) stehende der bisherige Listenplatz und nach dem Namen folgt das Ergebnis aus den Punkten/Stimmen.
Das Pgm stoppt - zur Verlesung der Ergebnisse - automatisch nach der Ausgabe von jeweils 5 Ergebnissen und kann dann durch Drücken irgendeiner Taste fortgesetzt werden.
Es erfolgt die Ausgabe eines kompletten Druckprotokolls.
Falls mehr als 99 Teilnehmer erfaßt werden sollen, sind die DIM-Statements in den Zeilen 110 - 120 - 130 entsprechend der Teilnehmerzahl + 1 zu erhöhen.
Das Programm läuft auf dem TI 99/4 A mit Extended-Basic-Modul + Drukker.

Herbert May

```
1 CALL SCREEN(16)
2 DIM LP(100)
100 DIM CY$(100)
110 DIM A(100)
120 FOR I=1 TO 100
121 PRINT
122 INPUT "LISTENPLATZ     ":LP(I
)
123 IF LP(I)=0 THEN 220
124 INPUT "NAME            ":CY$(
I)
125 INPUT "ERGEBNIS        ":A(I)
126 N=N+1
127 LP(I)=LP(I)/100
128 A(I)=A(I)+LP(I)
129 NEXT I
220 GOSUB 320
221 CALL CLEAR
230 FOR I=1 TO 100
240 AA=ABS(A(I)-INT(A(I)))
250 A(I)=INT(A(I))
260 AA=AA*100
270 B=B+1
275 PRINT : :
280 PRINT :B;".(";AA;")";TAB(12)
;CY$(AA);":";TAB(25);A(I)
285 OPEN #1:"TP.E",OUTPUT :: PRI
NT #1:B;".(";AA;")";TAB(12);CY$(
AA);":";TAB(25);A(I):: CLOSE #1
290 Z=Z+1
300 IF Z=5 THEN 590
301 IF AA=0 THEN 311
310 NEXT I
311 END
320 HL=N
330 FOR HK=INT(N/2)TO 1 STEP -1
340 HI=HK
350 GOSUB 460
360 NEXT HK
370 FOR HK=N TO 2 STEP -1
380 HH=A(1)
390 A(1)=A(HK)
400 A(HK)=HH
410 HI=1
420 HL=HK-1
430 GOSUB 460
440 NEXT HK
450 RETURN
460 HJ=2*HI
470 IF HJ>HL THEN 570
480 IF HJ=HL THEN 510
490 IF A(HJ)<A(HJ+1)THEN 510
500 HJ=HJ+1
510 IF A(HJ)>A(HI)THEN 570
520 HH=A(HI)
530 A(HI)=A(HJ)
540 A(HJ)=HH
550 HI=HJ
560 GOTO 460
570 RETURN
580 Z=Z+1
590 Z=0
600 GOSUB 620
610 GOTO 310
620 CALL KEY(0,J,ST)
630 IF ST=0 THEN 620
640 IF J=74 THEN 650
650 RETURN
660 Z=0
```

Kleinanzeigen

Biete an Software

■ SINCLAIR-ZUBEHÖR VOM FACHMANN ■

Hardware	DM
neu: ZX Spectrum 16K	650
neu: ZX Spectrum 48K	830
Seikosha GP-100 80 Zeichen	995
Große Tastatur	245
Memdpak 16K Erweiterung	165
Memopak 32K Erweiterung	265
Momopak 64K Erweiterung	365
HRG Hochauflösende Grafik	215
Memopak I/F Centronicsschn.	215
Soundbox 3-Fach-Tongener.	155
Software	**DM**
1 K Chess läuft auf 1 K	35
ZX Chess 2 läuft für 16 K	69
Toolkit Befehlserweiterung	49
Aszmic-Rom Assembl. Betr.	215
Puckmann Pacman-Version	40
Pilot Flugsimulator	40

■ Überw. an Dresdn. Bank, W'burg ■
■ Kto. 3294225, BLZ 79080052 ■
■ Info g. DM 2 a/Konto von hsd-soft, ■
■ Pf. 1418, CH-6020 Emmenbrücke ■

******* SUPER-VEW ***** Einmaliges EXBASIC-Level-II-Prg. zur Auswertung sämtlicher VEW-Systeme (6/49 & 7/38). Info: Th. Preymesser, Hesselbergring 19, 8500 Nürnberg 60**

TI99/4A SOFTWARE - z.B. Kleinbuchstaben, SPIELE, MATHE etc. ZX 81 PROGRAMME ab 1,- !!! H. RAFEINER 2 HAMBURG 54 LOKSTEDTER STEINDAMM 69A

VC20-ANWENDER * Programme direkt vom Autor. HAUSH.-KONTO-FÜHRUNG mit max. 50 Konten + 13 S. Handbuch nur 50 DM. Info gegen 2 DM in Briefmarken von Elmar Maronn, Rathaus-Str. 306, 7981 FRONHOFEN * PROFITWARE

ZX-SPECTRUM 16K NUR 496.- DM
ZX-SPECTRUM 48K NUR 724.- DM
SPECTRUM 32K-RAM NUR 198.- DM
NEU: Spectrum LICHT-GRIFFEL 16/48K m. Softw. NUR 89.- DM.
BÜCHER z.B. Complete ROM Disassembly 44.80 DM
Spec 3-D Spawn of Evil 29.80 DM
Spec 3-D Pharao's Tomb 34.80 DM
Spec Voice Chess, das sprechende Schachprogramm(48K) 49.80 DM
Spectral Invaders 24.80 DM
Spec FORTH, incl. Editor 88.80 DM
Aspect Editor/Assembler 44.80 DM
VC20 SCRAMBLE 34.80 DM
VC CHESS 34.80 DM
VC Games Pack 29.80 DM
Auch Progr. für ZX81, ACRON ATOM und BBC. Katalog gegen 1.80 DM in Briefm. T. Wagner, Softwareversand, Postfach 112243H, 8900 Augsburg

Super VC-20 Programme ● Kauf od. Tausch über 200 Pgm ● Spiele ● Toolkits kommerz. Tel 0721/594785

VC-20: Spielpr. günstigst; Info mit Spielbeschr. gegen 1,20 DM; Fink, Oberstimmerstr. 33, 8070 Ingolstadt

** VC-20 Superprogramme je 5,- DM
** über 300 Programme auf Cassette
** Info 1,- DM bei Peter Doffine
** Heugärten 9; 4763 Ense 5

VC-20/VC-64 Software! Etwa 500 Programme: Tel. 02222/3640

Verk. VC-20 Software. Liste geg. 80 Pf. in Briefm.! Thomas Baum, Wimmelstr. 2a, 3500 Kassel

**** Apple-Software **** Große Auswahl von Apple II Software. Bitte Liste anfordern. Zuschriften unter 8305/1

Kostenlos - ZX 81 Programme aller Art - Info: Ingewai 16, 2280 Keitum Christian Goltz!

VC-20 Assemblerprogramme äußerst preiswert. Auch Tausch. Info gg. Rückporto bei A. Sauren, Vennstr. 49, 5190 Stolberg. Es lohnt sich!

VC-20 oder ZX 81 Software! Katalog VC-20 4 DM, Liste ZX 1 DM. Schnell anfordern von: TREICHEL, Mittelbruchzeile 105-1, 1000 Berlin 51.

**** VC-20 Soft- u. Hardware ****
* z.B. 8K Speicher 95 DM. Schnell- *
* Lademodul für Recorder 120 DM *
* viele Spiele u. anderes Zubehör *
* Info gegen Freiumschlag oder *
* 1,30 DM in Briefmarken an *
*** N. Flesch, Scheideweg 63 g ***
*** 4650 Gelsenkirchen-Buer ***

Super Arcade u. Strategiespiele (SF/Fantasy/Wargames) für Apple/TRS-80/Atari/PET/IBM-PC auf Cas/Disk. Katalog für DM 1,- in Briefm. von: Fantastic Shop, Abt. H, Postf. 3026, 4000 Düsseldorf 1

Hard- u. Software (Programme, Module) für TI-99/4a, VC-20, 64, MZ-80 A zu Sonderpreisen, Info gg. DM 1: Boxler, Mittelfeldstr. 50, 732 Göpp. 8.

■■■ * VC-20 Supersoftware * ■■■
■ Action&Lift, Info: 99 Pf, J. Güttel ■
■■■ Kölner Str. 99, 5650 Solingen ■■■

**** VC-20 Software??? ***
* Spitzensoftware in großer *
* Auswahl bei: H & M Kohlen *
* Volkartstr. 36, 8000 München 19 *
* Info geg. 2,- DM in Briefm. *
*** oder Ihre Tauschliste. ***

ZX-81 (16K): 2 Brettspiele => Backgammon + Mühle auf Cassette zusammen nur 10 DM (Info gegen Freiumschlag). S. Wimmel, Thielenstr. 4, 2800 Bremen 1

ZX-81-Katalog DM 1: Programme auf CC z.B. IQ, Multimaze. Nur bei C. Goltz, Ingewai 16, 2280 Keitum.

Verkaufe meine Software für VC-20 Liste gegen Rp
B. Konecny
O'Briengasse 54/1, 1210 Wien

Biete an Hardware

TI-99/4A, Joysticks, Extended ● Basic, Software, VB: 1050.- DM ● P. DASSOW, Tel. (0661) 44194 Sa! ●

Günstig! ZX81 MIT 16K ERWEITERUNG, KOFFER, PROGRAMMEN AUF KASSETTEN UND LISTEN, SPIELEBUCH GESAMT NUR 300.- DM. J. GROSSMANN NACHTIGALLENWEG 30, 2070 AHRENSBURG

- VC-20/VC-64 Kassetteninterface - jeder Recorder o. Tonbandgerät anschließbar. Fertiggerät im Geh. mit Steck. 48 DM auch für CBM. U-Portst. 9,80 DM Griff 3 DM NN Portofrei Fa. BOI, Hoye 1, 2225 Schafstedt, Tel. 04805/380.

Verkaufe ZX-81 + Memotech-16K + div. Software (Schach u. a.) + Bücher für 400 DM
Tel.: (0751) 93170

VC-64 + 1541 + Datasette 8 Wochen alt, 2000 DM Tel. 09602/5252 ab 18.00 Uhr

MZ-80 K mit Deb., ML, ASS, Extend.-Basic, Pascal König (07159) 6779

Suche Software

Suche Pascal für Apple II Euro + und andere Sprachen. M. Köwing Lohbrügger Weg 1, 2050 Hamb. 80

Suche Software für alle Homecomputer. Nehme alle Pgm. C. Zalesiak, Broicher Str. 42, 4200 Oberhausen 1

Verschiedenes

Hochaufl. Grafik f. Epson MX-80 FT Umbau in 3 Min. Statt 298 nur 98 DM, Info oder Scheck an E. Noack, Wilhelmstr. 43, 4620 Castrop-R.

● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
● **ZX-81-Monitoranschluß:** ●
● Zeichen grün auf schwarz; auch ●
● Invertierung d. Fernsehbildes ●
● möglich. Info: F. Kopitzki, ●
● Ameisenbergstr. 57 c, 7 Stgt. 1 ●
● ● ● ● ● ● ● ● ● ● ● ● ● ● ●

Kontakte

Suche Kontakt zu VC-20 Besitzern. Robert Elmshäuser, Teichweg 6 3550 Marburg 7

Freie Mitarbeiter für Softwareentwicklung VC-20/64 gesucht. Assemblerkenntnisse erforderlich. Tel.: 06121/708427, 8-16 Uhr.

Nutzen Sie den Kleinanzeigenteil.

Homecomputer ist die Zeitschrift für die Anwendung der kleinen Microcomputer. Im Heim, in der Schule, im Club. Zum Spielen, Lernen und Verwalten.

Fortsetzung von Seite 7

```
4660 PRINT "BITTE DRUECKEN:
                J >>> NOCH EIN
SPIEL           N >>> ENDE"
4670 CALL KEY(0,KEY,STATUS)
4680 IF STATUS=0 THEN 4670
4690 IF KEY=106 THEN 390
4700 IF KEY=74 THEN 390
4710 IF KEY=78 THEN 4730
4720 IF KEY=110 THEN 4730 EL
SE 4670
4730 CALL CLEAR
4740 END
```

Space-Defence TI-99/4A

Ein weiteres Spiel, dessen Schauplatz das Weltall ist.

Anfang: Die Zahlen hinter den symbilisierten Raumbasen geben die Nummern derselben an. Die blinkende Raumbasis markiert die jeweilige Position. Die Anzahl der gegnerischen Raumschiffe wird angezeigt. Fliegen Sie nun zur Raumbasis Ihrer Wahl durch Eintasten der entsprechenden Nummer.

Flug: Sie können verschiedene Geschwindigkeiten wählen, die Sie verschieden schnell ans Ziel führen und einen verschieden großen Energieverbrauch zur Folge haben. Die größte Geschwindigkeit ist die Lichtgeschwindigkeit (Stufe 1), die kleinste 1/9 LG (Stufe 9) - Tasten 1 - 9.

Während des Fluges können Angreifer entgegenkommen, die als Abfangjäger fungieren und Sie treffen sollen!

Je nach Geschwindigkeit verursachen sie unterschiedlich großen Schaden an Ihrem Schiff (größte Geschwindigkeit - kleinster Schaden) der nach 96 % zum Tode führt. Dieser Schaden kann aber nur während des Fluges oder durch Andocken an die Raumbasis verhindert werden. Durch Drücken der Taste B während des Fluges wird der Schaden unter Energieverlust repariert. Sie gelangen nicht eher an eine Basis, bevor Sie den Angreifer abgeschossen haben.

Dies geht so: Mit dem Joystick ist der Gegner ins Fadenkreuz zu lenken und mit der Taste abzuschießen.

Zu Ihrem Schutz können Sie ein Schutzschild errichten, was aber 5% Energie kostet. Außerdem stehen zur Abwehr 9 Laserschüsse zur Verfügung. Wollen Sie schneller ans Ziel, so können Sie mit Hyperspace fliegen (Taste H), was wiederum erhöhten Energieverbrauch zur Folge hat.

Sind keine Angreifer vorhanden, so können Sie Ihr Raumschiff das unten rechts sichtbar wird mittels Joystick an die Basis andocken und Energie nachtanken. In dieser Zeit werden gleichzeitig Schäden am Schiff beseitigt. Haben Sie genug getankt, so können Sie mit der G-Taste die Verteilung der Angreifer an den jeweiligen Basen wieder sichtbar machen.

Während des gesamten Spieles versuchen die Angreifer ihre Zahl zu erhöhen, was nur durch Ihre Angriffe verhindert werden kann. Während Sie angreifen, können Sie den Beschuß Ihrer Raumbasis beobachten und sich im Falle eines Treffers ärgern, daß Sie sie nicht abschießen konnten.

Tastenbelegung:

S = Schutzschild
R = Reparatur
G = Raumbasenverteilung
1-9= Geschwindigkeiten
J = Neues Spiel
H = Hyperspace

Ursachen für den Verlust Ihres Raumschiffes:

zu großer Schaden
keine Energie
keine Raumbasen
keine Laser, demzufolge zu großer Schaden

```
10  CALL  CLEAR
20  RANDOMIZE
30  CALL  CHAR(91,"3C4299A1A19
942 3C")
40  AB$="0404SPACE-DEFENCE II
"
50  GOSUB  90
60  AB$="0804[ Nov.  '82 BY PR
L"
70  GOSUB  90
80  GOTO  3730
90  ZQ=VAL(SEG$(AB$,1,2))
100  SQ=VAL(SEG$(AB$,3,2))
110  LQ=LEN(AB$)-4
120  AB$=SEG$(AB$,5,LQ)
130  FOR  XQ=SQ  TO  SQ+LQ-1
140  CALL  VCHAR(ZQ,XQ,ASC(SEG
$(AB$,XQ-SQ+1,1)))
150  NEXT  XQ
160  RETURN
170  AD=AD+SGN(AD-4)*-15
180  CALL  COLOR(13,AD,1)
190  CALL  COLOR(14,AD+SGN(AD-
4)*-15,1)
200  RETURN
210  X=INT(RND*6+1)
220  AZ(X)=AZ(X)+RND/3+(AZ(X)
>8)*(AZ(X)-8)
230  RETURN
240  CALL  HCHAR(J(2),J(1),32)
250  CALL  CHAR(116,"00000000F
F")
260  CALL  HCHAR(ZL+3,ST+6,116
)
270  J(2)=23
280  J(1)=28
290  CALL  HCHAR(J(2),J(1),90)
300  CALL  HCHAR(J(2),J(1)+1,9
1)
310  IF  (J(1)=ST+7)+(J(2)=ZL+
3)=-2  THEN  440
320  CALL  JOYST(1,J1,J2)
330  ON  (RND<.8)+2  GOSUB  210,
170
340  IF  J1-J2+J1=0  THEN  320
350  JC1=J(1)
360  JC2=J(2)
370  J(1)=INT(31*((J(1)+J1/4-
1)/31-INT((J(1)+J1/4-1)/31))
)+1
380  J(2)=INT(23*((J(2)-J2/4-
1)/23-INT((J(2)-J2/4-1)/23))
)+1
390  CALL  HCHAR(JC2,JC1,0)
400  CALL  HCHAR(JC2,JC1+1,01)
410  CALL  GCHAR(J(2),J(1),0)
420  CALL  GCHAR(J(2),J(1)+1,0
1)
430  GOTO  290
440  E=9
450  CALL  SOUND(-100,-3,0,110
,5)
460  CALL  HCHAR(24,30,57)
470  CALL  SOUND(-1000,-3,5,40
0,9)
480  CALL  COLOR(12,8,5)
490  EN=EN+3+(EN>96)*3
500  GOSUB  3690
510  SC=SC-3-(SC<3)*3
520  GOSUB  3340
530  CALL  KEY(3,K,X)
540  CALL  SOUND(-1000,-2,7,44
0,9)
550  CALL  COLOR(12,8,2)
560  ON  (RND<.7)+2  GOSUB  210,
170
570  IF  K=71  THEN  790 ELSE 47
0
580  CALL  JOYST(1,J1,J2)
590  CALL  KEY(1,K,X)
600  X=4-ABS(X)*2
610  IF  J1-J2+J1=0  THEN  780
620  CALL  HCHAR(J(2),J(1),0)
630  J(1)=INT(32*((J(1)+J1/X-
1)/32-INT((J(1)+J1/X-1)/32))
)+1
640  J(2)=INT(24*((J(2)-J2/X-
1)/24-INT((J(2)-J2/X-1)/24))
)+1
650  CALL  GCHAR(J(2),J(1),0)
660  CALL  HCHAR(J(2),J(1),116
)
670  IF  0<>152  THEN  780
680  U=32
690  CALL  SOUND(900,-6,0)
700  V=V+100
710  AZ(POI)=INT(AZ(POI)-1)
720  FOR  X=1  TO  CPO
730  CALL  SOUND(-300,-3,3)
740  IF  (PN(X,1)=J(2))+(PN(X,
2)=J(1))=-2  THEN  770
750  CALL  SOUND(-300,-7,0,333
,1)
```

```
760  NEXT X
770  PN(X,1)=0
780  RETURN
790  CALL CHAR(117,"FF187EFFF
F7E18FF")
800  CALL HCHAR(2,1,32,704)
810  CALL COLOR(11,8,1)
820  IF RABA(1)+RABA(2)+RABA(
3)+RABA(4)+RABA(5)+RABA(6)<>
35 THEN 850
830  AB$="0404ALLE@RAUMBASEN@
VERLUREN"
840  GOTO 3420
850  RESTORE 960
860  FOR X=1 TO 6
870  AZ(X)=ABS(INT(AZ(X)+(AZ(
X)>=9)*(AZ(X)-8)))
880  READ ST,ZL
890  IF POI<>X THEN 920
900  AA=ZL
910  A=ST
920  CALL HCHAR(ZL,ST,117)
930  CALL HCHAR(ZL+1,ST,152,A
Z(X))
940  CALL HCHAR(ZL,ST-2,X+48)
950  NEXT X
960  DATA 5,5,20,3,16,16,22,1
8,10,20,24,22
970  CALL HCHAR(AA,A,42-RABA(
POI))
980  CALL KEY(3,X,S)
990  IF (X<49)+(X>54)=0 THEN
1040
1000  CALL HCHAR(AA,A,117)
1010  CALL SOUND(80,440,4)
1020  GOSUB 210
1030  GOTO 970
1040  CALL SOUND(440,440,1)
1050  X=X-48
1060  W=EG(POI,X)
1070  IF (W=0)+(RABA(X)>0)THE
N 970
1080  POI=X
1090  GOSUB 2190
1100  B=(RND*6+1)*14
1110  CALL SOUND(-3000,-6,14)
1120  CALL HCHAR(2,1,32,704)
1130  CALL COLOR(14,16,1)
1140  FOR X=128 TO 143
1150  CALL HCHAR(RND*18+3,RND
*26+5,X)
1160  NEXT X
1170  CALL HCHAR(24,22,45)
1180  CALL SOUND(-4250,-6,14)
1190  RESTORE 1260
1200  RESTORE 1250
1210  FOR X=1 TO 10
1220  READ A,AB$
1230  CALL CHAR(A,AB$)
1240  NEXT X
1250  DATA 121,FFFFFFFFFFFFF07
03,120,FFFFFFFFFFFFE0C0,122,
C0E0FFFFFFFFFFFF,123,0307FFF
FFFFFFFFFF,152,FF993C3C99FF
1260  DATA 115,0040060E1CF8F0
00,114,0000000000FFFF,113,F0
E0C0C0FFFF,112,0F07030303F
FFF,104,0080FFFFFFFFFFFF
1270  GOSUB 170
1280  DATA 000F1F387060,00FFF
F,00FFFF0303030F0F,00FFFFC0C
0C0E2F0,00FFFF,00F0F81C0E06,
000007 0F1F3F7F7F
1290  DATA 4001FFFFFFFFFFFF,0
080FFFFFFFFFFF,0000E0F0F8FC
FEFE,7F7F3F1F0F07
1300  DATA FFFFFFFFFFFF01,FFF
FFFFFFF80,FEFEFCF8F0E0
1310  DATA 000460703 81F0F,000
00000 00FFFF
1320  CALL COLOR(13,16,1)
1330  CALL HCHAR(23,28,90)
1340  CALL HCHAR(23,29,91)
1350  ZL=INT(RND*12+4)
1360  ST=INT(RND*16+8)
1370  CALL SOUND(-4250,-6,14)
1380  RESTORE 1470
1390  FOR X=0 TO 3
1400  CALL COLOR(X+9,1,1)
1410  GOSUB 170
1420  FOR A=0 TO 5
1430  READ AA
1440  CALL HCHAR(ZL+X,ST+A,AA
)
1450  NEXT A
1460  NEXT X
1470  DATA 96,97,98,99,100,10
1,102,103,120,121,104,105,10
6,107,122,123,108,109,110,11
1,112,113,114,115
1480  CALL COLOR(9,8,1)
1490  CALL COLOR(12,8,5)
1500  CALL COLOR(10,8,1)
1510  CALL COLOR(11,8,1)
1520  CPO=AZ(POI)
1530  CALL CHAR(116,"817E665A
5A66,'E81")
1540  CALL COLOR(16,14,1)
1550  FOR X=1 TO AZ(POI)
1560  PN(X,1)=INT(RND*20+3)
1570  IF (PN(X,1)>ZL-2)+(PN(X
,1)<ZL+7)=-2 THEN 1560
1580  PN(X,2)=INT(RND*26+5)
1590  IF (PN(X,2)>ST-2)+(PN(X
,2)<ST+5)=-2 THEN 1580
1600  CALL HCHAR(PN(X,1),PN(X
,2),152)
1610  PN(X,3)=ZL+2+SGN(ZL-PN(
X,1))*-3
1620  PN(X,4)=ST+3+SGN(ST-PN(
X,2))*-4
1630  NEXT X
1640  J(2)=20
1650  J(1)=28
1660  CALL HCHAR(J(2),J(1),11
6)
1670  AZ(POI)=INT(AZ(POI))
1680  A=INT(RND*CPO+1)
1690  IF (PN(A,1)=0)+(AZ(POI)
<=0)THEN 1740
1700  CALL HCHAR(PN(A,1),PN(A
,2),32)
1710  PN(A,1)=PN(A,1)+SGN(PN(
A,3)-PN(A,1))
1720  PN(A,2)=PN(A,2)+SGN(PN(
A,4)-PN(A,2))
1730  CALL HCHAR(PN(A,1),PN(A
,2),152)
1740  IF RND<.7 THEN 1760
1750  GOSUB 170
1760  GOSUB 580
1770  IF (RND>.2*AZ(POI))+(AZ
(POI)<1)THEN 1820
1780  GOSUB 210
1790  IF RND<.5 THEN 1940
1800  IF RND<.1 THEN 1860
1810  GOSUB 3280
1820  CALL KEY(3,K,S)
1830  ON -(K=83)-2*(K=71)-3*(
((K=65)+(AZ(POI)<=0))=-2)+1
GOTO 1670,1840,790,240
1840  GOSUB 3650
1850  GOTO 1670
1860  CALL SCREEN(16)
1870  CALL SOUND(400,-3,1,600
,0)
1880  CALL SCREEN(2)
1890  CALL HCHAR(J(2),J(1),0)
1900  J(2)=23
1910  J(1)=26
1920  CALL HCHAR(J(2),J(1),11
6)
1930  GOTO 1670
1940  CALL COLOR(9,2,7)
1950  CALL SOUND(-20,-3,0)
1960  CALL COLOR(10,2,7)
1970  CALL SOUND(-20,-7,0)
1980  CALL COLOR(11,2,7)
1990  CALL SOUND(-20,-3,0)
2000  CALL COLOR(9,8,1)
2010  CALL SOUND(-20,-7,0)
2020  CALL COLOR(10,8,1)
2030  CALL SOUND(-20,-3,0)
2040  CALL COLOR(11,8,1)
2050  IF RND>.06 THEN 1670
2060  FOR X=400 TO 110 STEP -
5
2070  CALL SOUND(-90,X,0,X+20
,1,X+30,2)
2080  NEXT X
2090  FOR X=9 TO 12
2100  CALL SOUND(400,-5,(12-X
)*2,660,6)
2110  CALL COLOR(X,2,2)
2120  NEXT X
2130  RABA(POI)=7
2140  AB$="0403RAUMBASIS@"&ST
R$(POI)&"@SCHROTT@"
2150  GOSUB 90
2160  FOR X=1 TO 200
2170  NEXT X
2180  GOTO 790
2190  CALL HCHAR(2,1,32,704)
2200  CALL COLOR(8,16,1)
2210  CALL COLOR(10,7,1)
2220  FOR X=11 TO 16
2230  CALL COLOR(X,1,1)
2240  NEXT X
2250  IF Z>0 THEN 2270
2260  PRINT "ENERGIE:99;@ENER
GIESCHILD:@@"::::::::::::::::
::::::::::"HAVARIE:0@;@@LGF:1/
9@LASER:9";
2270  GOSUB 3590
2280  Z=11
2290  FOR A=112 TO 158 STEP 8
2300  FOR AA=A TO A+7
```

```
2310 CALL HCHAR(18*RND+4,25*
RND+4,AA)
2320 NEXT AA
2330 CALL HCHAR(12,16,67-A/8
)
2340 CALL SOUND(200,600,3)
2350 NEXT A
2360 CALL HCHAR(12,16,32)
2370 CALL SOUND(900,-6,16)
2380 CALL HCHAR(12,3,88,8)
2390 CALL HCHAR(12,22,88,8)
2400 CALL VCHAR(3,16,89,6)
2410 CALL VCHAR(16,16,89,6)
2420 B=112
2430 Z=Z+1
2440 EN=EN-(128-B)/200
2450 W=W-10+B/14
2460 IF (W>0)+(CO>0)=0 THEN
2770
2480 IF Z<17 THEN 2570
2490 GOSUB 3690
2500 GOSUB 210
2510 X INT(RND*6)+1
2520 IF (RND>AZ(X)*.01)+(X=P
OI)+(RABA(X)=7)THEN 2560
2530 RABA(X)=7
2540 AB$="0403RAUMBASIS@"&ST
R$(X)&"@SCHROTT"
2550 GOSUB 90
2560 Z=11
2570 CALL COLOR(Z,INT(RND*14
+3),1)
2580 CALL KEY(3,K,S)
2590 IF S=0 THEN 2840
2600 IF (K<>82)+(SC<1)THEN 2
650
2610 SC=INT(SC-SC/10)
2620 EN=INT(EN-SC/12)
2630 GOSUB 3340
2640 GOSUB 3690
2650 IF (K<>72)+(CO=1)THEN 2
780
2660 FOR X=1 TO W/80
2670 FOR Z=11 TO 16
2680 CALL HCHAR(12,16,72)
2690 CALL COLOR(Z,INT(RND*16
+1),1)
2700 CALL SOUND(-999,-2,2*(1
6-Z),400,3*(16-Z))
2710 CALL HCHAR(12,16,32)
2720 CALL COLOR(Z,1,1)
2730 NEXT Z
2740 NEXT X
2750 EN=EN-6*W/80
2760 GOSUB 3690
2770 RETURN
2780 IF K<>83 THEN 2810
2790 GOSUB 3650
2800 GOTO 2840
2810 IF (K<49)+(K>57)THEN 28
40
2820 B=(K-49)*14
2830 CALL HCHAR(24,22,K)
2840 CALL COLOR(Z,1,1)
2850 FOR A=1 TO B
2860 NEXT A
2870 IF (RND*9<.5)+(CO<>1)<>
2 THEN 2950
2880 PY=INT(RND*22+2)
2890 PX=INT(RND*30+2)
2900 H=7
2910 CALL CHAR(104,A$(7))
2920 CALL SOUND(-800,440,0,-
6,6)
2930 CO=1
2940 GOTO 3240
2950 IF CO=0 THEN 3260
2960 IF H=4 THEN 2990
2970 H=H-1
2980 CALL CHAR(104,A$(H))
2990 IF RND>.1 THEN 3010
3000 GOSUB 3280
3010 CALL JOYST(1,DX,DY)
3020 CALL KEY(1,K,S)
3030 IF (K<>18)+(E=0)THEN 32
00
3040 CALL HCHAR(12,11,88,11)
3050 CALL VCHAR(9,16,89,7)
3060 CALL SOUND(-200,-3,0)
3070 CALL HCHAR(12,11,32,11)
3080 CALL VCHAR(9,16,32,7)
3090 E=E-1
3100 CALL HCHAR(24,30,E+48)
3110 IF (PY=12)+(PX=16)<>-2
THEN 3190
3120 V=V+1000
3130 CALL HCHAR(PY,PX,32)
3140 CALL SOUND(-400,-7,2)
3150 CALL SOUND(100,-7,0)
3160 AZ(POI)=ABS(AZ(POI)-1)
3170 CO=0
3180 GOTO 3260
3190 CALL HCHAR(PY,PX,104)
3200 IF DX-DY+DX=0 THEN 3260
3210 CALL HCHAR(PY,PX,HH)
3220 PX=INT(32*((PX-DX/4-1)/
32-INT((PX-DX/4-1)/32)))+1
3230 PY=INT(24*((PY+DY/4-1)/
24-INT((PY+DY/4-1)/24)))+1
3240 CALL GCHAR(PY,PX,HH)
3250 CALL HCHAR(PY,PX,104)
3260 CALL SOUND(-4250,-6,16-
((Z-11)*2))
3270 GOTO 2430
3280 FOR X=1 TO RND*3+3
3290 CALL SCREEN(6)
3300 CALL SOUND(-100,-7,12-2
*T)
3310 CALL SCREEN(2)
3320 NEXT X
3330 SC=SC+INT(RND*(10-SS*5+
B/3))
3340 CALL HCHAR(24,11,ASC(ST
R$(SC)))
3350 CALL HCHAR(24,12,ASC(SE
G$(STR$(SC)&"@",2,1)))
3360 IF SC>95 THEN 3410
3370 SS=0
3380 CALL COLOR(8,16,1)
3390 CALL HCHAR(1,29,64)
3400 RETURN
3410 AB$="0403"&STR$(SC)&";@
HAYARIE@:>SCHROTT."
3420 CALL SOUND(-500,440,0)
3430 GOSUB 90
3440 AB$="0603@"&STR$(V)&"@P
UNKTE@ @NOCHMAL@(J)"
3450 GOSUB 90
3460 CALL KEY(3,K,S)
3470 CALL SOUND(20,660,3)
3480 IF S=0 THEN 3460
3490 FOR X=1 TO 999
3500 NEXT X
3510 CALL HCHAR(1,1,32,768)
3520 IF K=74 THEN 4250
3530 STOP
3540 AB$="0403@KEINE@ENERGIE
@MEHR"
3550 GOTO 3420
3560 FOR A=112 TO 151
3570 CALL CHAR(A,A$(INT(RND*
3+1)))
3580 NEXT A
3590 FOR X=112 TO 158
3600 IF X<>124 THEN 3620
3610 X=152
3620 CALL CHAR(X,A$(INT(RND*
3+1)))
3630 NEXT X
3640 RETURN
3650 EN=EN-5
3660 CALL HCHAR(1,29,42)
3670 CALL COLOR(8,5,1)
3680 SS=1
3690 CALL HCHAR(1,11,INT(EN/
10)+48)
3700 CALL HCHAR(1,12,ASC(SEG
$(STR$(INT(EN))&"@",2,1)))
3710 IF EN<1 THEN 3540 ELSE
3640
3730 RESTORE 3780
3740 FOR X=1 TO 7
3750 READ A$(X)
3770 NEXT X
3780 DATA 1034DE6CFA3C18,181
8,3C3C,C399A5E7A599C3,00425A
7E5A42,0000243C24,00001818
3790 RESTORE 3860
3800 FOR X=1 TO 6
3810 FOR A=1 TO 6
3820 READ AA
3830 EG(X,A)=AA*100
3840 NEXT A
3850 NEXT X
3860 DATA 0,4,4,6,4,7,4,0,3,
3,5,6,4,4,3,0,2,2,2,6,3,5,2,
0,2,5,1,4,6,2,2,5,0,2,7,4,2,
1,2,0
3870 RESTORE 3930
3880 CALL SCREEN(2)
3890 FOR X=1 TO 6
3900 READ S,S$
3910 CALL CHAR(S,S$)
3920 NEXT X
3930 DATA 88,000000FF,89,080
808080808080808,64,,59,0062640
810264600,48,0038444C5464443
8,91,00C0F0F8FFF87060
3940 RESTORE 1310
3950 RESTORE 1300
3960 RESTORE 1290
3970 RESTORE 1280
3980 FOR X=96 TO 111
3990 READ B$
4000 CALL CHAR(X,B$)
4010 NEXT X
4020 GOSUB 3560
```

```
4030  CALL CLEAR
4040  FOR X=2 TO 8
4050  CALL COLOR(X,8,1)
4060  NEXT X
4070  FOR A=128 TO 158
4080  CALL HCHAR(RND*22+1,RND
*30+1,A)
4090  NEXT A
4100  AB$="0404DAS WELTALL -"
4110  GOSUB 90
4120  AB$="06086RENZE ZUM UNE
NDLICHEN"
4130  GOSUB 90
4140  FOR A=1 TO 20
4150  AA=INT(RND*4+13)
4160  FOR X=1 TO 35
4170  NEXT X
4180  CALL COLOR(AA,16,1)
4190  FOR X=1 TO 35
4200  NEXT X
4210  CALL COLOR(AA,1,1)
4220  NEXT A
4230  CALL CLEAR
4240  CALL CHAR(90,"00030F1FF
F1F0F06")
4250  AD=1
4260  E=9
4270  EN=99
4280  TE=100
4290  Y=0
4300  ZL=4
4310  ST=4
4320  SC=0
4330  Z=0
4340  FOR X=1 TO 6
4350  CALL COLOR(X+1,16,3)
4360  AZ(X)=INT(RND*7)
4370  RABA(X)=0
4380  NEXT X
4390  PUT=INT(RND*6)+1
4400  GOTO 790
```

Street Race

ein Spielprogramm für den TI 99/4A in der Grundversion (also ohne Extended Basic Modul). Das Pgm nennt sich STREET RACE und besteht darin, daß der Spieler möglichst vielen Hindernissen auf einer Straße ausweichen soll, z.B. Absperrungen, Schlaglöcher etc.

Durch Druck auf eine Taste kann jeweils von der linken auf die rechte Spur (und umgekehrt) gewechselt werden. Der Wagen des Spielers, angedeutet durch einen blauen Pfeil in Fahrtrichtung, bleibt am oberen Bildschirmrand, während die Straße (samt Hindernissen) sich auf dem Bildschirm langsam nach oben schiebt. Die Spielanleitung ist in Kurzfassung auch im Pgm enthalten. Nach der Wahl der Startposition (links/rechts) dauert es etwa 30 Sekunden bis das Spiel gestartet wird. Während dieser Zeit erscheint auf dem Bildschirm „Bitte warten", und der Computer speichert intern die Verteilung der Hindernisse. Hat man die ersten 100 Streckeneinheiten (sprich Druckzeilen) zurückgelegt, so erscheint eine Meldung und der Computer erhöht den Schwierigkeitsgrad. Schafft man auch diesmal 100 Zeilen, wird der Schwierigkeitsgrad noch einmal erhöht. Bei diesem Grad bleibt's dann; aber es ist sowieso fast unmöglich, ein drittes Mal 100 Zeilen heil zu überstehen. Bei einem Zusammenstoß druckt das Pgm die erreichte Punktzahl, das bisherige Höchstergebnis und die Frage nach einem weiteren Spiel aus.

Zum Aufbau des Pgm's in Stichpunkten:

Zeilen	Funktion
100-200	Definition und Einfärben der neuen Zeichen
230-420	Einleitung, Festlegung der Startposition u. d. Schwierigkeitsgrades
430-730	Streckenfestlegung. Hindernisse per Zufallszahlen
740-930	Streckenbezeichnung, Steuerung, Feststellung, ob Hindernis
960-1120	Meldung: Zusammenstoß. Frage nach neuem Spiel
1130-1230	Meldung: eine Runde beendet. Erhöhung des Schwierigkeitsgrades, sofern nicht bereits Grad 3.
1240+1250	Ende.

Anbei das Programmlisting, in Ermangelung eines Druckers mit Schreibmaschine getippt.
Noch eine Anmerkung zum Pgm:
Das Festlegen der Hindernisse in den Zeilen 510-730 wäre mit DATA-Zeilen programmäßig einfacher zu bewerkstelligen gewesen, hätte aber wegen der READ-Befehle, die ihrerseits in FOR-NEXT-Schleifen eingebaut hätten werden müssen, die Zeit zur Hindernisfestlegung enorm verlängert. So beträgt sie nur ca. 25 Sekunden.

Alwin Ertl

```
10 REM   *************
20 REM   *STREET RACE*
30 REM   *************
40 REM   *C BY A.    *
50 REM   *ERTL, 1983 *
60 REM   *************
100 RANDOMIZE
110 CALL CHAR(130,"383838383
8783 81")
120 CALL CHAR(140,"183C7EFFF
F7E3C18")
130 CALL CHAR(150,"000000040F
F4")
140 CALL CHAR(151,"00000002F
F02")
150 CALL CHAR(155,"030703E1C3
8703E0C"
160 CALL CHAR(120,"FFFFFFFFF
FFFFFF")
170 CALL COLOR(12,15,15)
180 CALL COLOR(13,5,15)
190 CALL COLOR(14,2,15)
200 CALL COLOR(15,14,15)
210 CALL COLOR(16,10,16)
220 CALL CLEAR
230 PRINT "+++++++++++++":"+
    STREET RACE+":"+++++++++
    ++++":::::::::::::::::
```

```
24Ø PRINT "BENOETIGEN SIE DI
    E":"SPIELREGELN? /J/ /N/
    "
25Ø CALL KEY(Ø,K,S)
26Ø IF S=Ø THEN 25Ø
27Ø IF K=74 THEN 31Ø
28Ø IF K=78 THEN 35Ø
29Ø CALL SOUND(2ØØ,-1,Ø)
3ØØ GOTO 25Ø
31Ø CALL CLEAR
32Ø PRINT "SIE SOLLEN EINEN
    WAGEN,":"DURCH ";CHR$(13
    Ø);" ANGEDEUTET,"
33Ø PRINT "DURCH EINEN HINDE
    RNISKURS":"MANOEVRIEREN!
    DURCH DRUECKEN"
34Ø PRINT "EINER TASTE WECHS
    ELN SIE":"DIE FAHRSPUR (
    LINKS/RECHTS)."
35Ø PRINT ::::"ALLES KLAR?":
    ::"AUF WELCHER FAHRSPUR"
    :"MOECHTEN SIE BEGINNEN,
    "
36Ø INPUT "LINKS ODER RECHTS
    ? ":A$
37Ø IF SEG$(A$,1,1)="R" THEN
    4ØØ
38Ø A=14
39Ø GOTO 41Ø
4ØØ A=19
41Ø P=Ø
42Ø S=3
43Ø DIM S$(1ØØ)
44Ø CALL CLEAR
45Ø PRINT TAB(5);"BITTE WART
    EN!"::::::::::
46Ø A$="          "
47Ø B$=CHR$(12Ø)&CHR$(12Ø)&C
    HR$(12Ø)&CHR$(12Ø)&CHR$(
    12Ø)
48Ø FOR I=1 TO 1ØØ
49Ø IF I/S<>INT(I/S) THEN 54
    Ø
5ØØ X=INT(9*RND)+1
51Ø ON X GOSUB 56Ø,58Ø,6ØØ,6
    2Ø,64Ø,66Ø,68Ø,7ØØ,72Ø
52Ø NEXT I
53Ø GOTO 74Ø
54Ø S$(I)=A$&B$&B$&A$
55Ø GOTO 52Ø
56Ø S$(I)=A$&CHR$(12Ø)&CHR$(
    12Ø)&CHR$(12Ø)&CHR$(14Ø)
    &CHR$(12Ø)&B$&A$
57Ø RETURN
58Ø S$(I)=A$&CHR$(12Ø)&CHR$(
    12Ø)&CHR$(14Ø)&CHR$(12Ø)
    &CHR$(12Ø)&b$&A$
59Ø RETURN
6ØØ S$(I)=A$&B$&CHR$(12Ø)&CH
    R$(12Ø)&CHR$(14Ø)&CHR$(1
    2Ø)&CHR$(12Ø)&A$
61Ø RETURN
62Ø S$(I)=A$&CHR$(12Ø)&CHR$(
    12Ø)&CHR$(151)&CHR$(12Ø)
    &CHR$(12Ø)&B$&A$
63Ø RETURN
64Ø S$(I)=A$&B$&CHR$(12Ø)&CH
    R$(12Ø)&CHR$(15Ø)&CHR$(1
    2Ø)&CHR$(12Ø)&A$
65Ø RETURN
66Ø S$(I)=A$&CHR$(155)&CHR$(
    155)&CHR$(155)&CHR$(155)
    &CHR$(155)&B$&A$
67Ø RETURN
68Ø S$(I)=A$&B$&CHR$(155)&CH
    R$(155)&CHR$(155)&CHR$(1
    55)&CHR$(155)&A$
69Ø RETURN
7ØØ S$(I)=A$&B$&"STOP"&CHR$(
    12Ø)&A$
71Ø RETURN
72Ø S$(I)=A$&CHR$(12Ø)&"STOP
    "&B$&A$
73Ø RETURN
74Ø CALL SOUND(5ØØ,-3,5)
75Ø C$=A$&B$&B$&A$
76Ø FOR I=1 TO 24
77Ø PRINT C$
78Ø NEXT I
79Ø Z=Ø
8ØØ CALL HCHAR(1,A,13Ø)
81Ø CALL KEY(Ø,K,ST)
82Ø CALL SOUND(-5ØØ,12Ø,5,-7
    ,7)
83Ø IF ST=Ø THEN 88Ø
84Ø IF A=14 THEN 87Ø
85Ø A=14
86Ø GOTO 88Ø
87Ø A=19
88Ø CALL GCHAR(2,A,PIC)
89Ø IF PIC<>12Ø THEN 94Ø
9ØØ P=P+1
91Ø Z=Z+1
92Ø PRINT S$(Z)
93Ø IF Z=1ØØ THEN 113Ø ELSE
    8ØØ
94Ø CALL SOUND(2ØØ,-6,Ø)
95Ø CALL CLEAR
96Ø PRINT TAB(5); ZUSAMMENST
    OSS!"::::
97Ø PRINT "PUNKTZAHL:";P
98Ø IF P<=H THEN 1Ø5Ø
99Ø H=P
1ØØØ FOR SI=1 TO 5
1Ø1Ø FOR SJ=-3 TO -1
1Ø2Ø CALL SOUND(15Ø,SJ,3)
1Ø3Ø NEXT SJ
1Ø4Ø NEXT SI
1Ø5Ø PRINT "HIGHSCORE:";H:::
    ::
1Ø6Ø PRINT "NOCH EIN SPIEL?
    /J/ /N/"
1Ø7Ø CALL KEY(Ø,K,S)
1Ø8Ø IF S-Ø THEN 1Ø7Ø
1Ø9Ø IF K=74 THEN 35Ø
11ØØ IF K=78 THEN 124Ø
111Ø CALL SOUND(2ØØ,-1,Ø)
112Ø GOTO 1Ø7Ø
113Ø CALL CLEAR
114Ø PRINT TAB(4);"GRATULATI
    ON!"::
115Ø PRINT "SIE HABEN DIE ST
    RECKE MIT":"SCHWIERIGKE
    ITSGRAD";ABS(S-4)
116Ø PRINT "BEWAELTIGT.":::
117Ø S=S-1
118Ø IF S<Ø THEN 12ØØ
119Ø S=1
12ØØ PRINT "ES GEHT WEITER M
    IT":"STUFE";ABS(S-4);"!
    ":::::
121Ø FOR V=1 TO 15ØØ
122Ø NEXT V
123Ø GOTO 44Ø
124Ø CALL SOUND(5ØØ,88Ø,1)
125Ø END
```

Breakpoint TI-99/4A

Mit den Tasten Komma und Punkt kann man den schwarzen Schläger nach links und rechts bewegen. Mit diesem Schläger muß man verhindern, daß die rote Kugel aus dem Spielfeld fliegt. Ziel des Spieles ist es, alle Blauen Punkte mit der Kugel abzuräumen. Wenn kein blauer Punkt mehr da ist, dann muß man die Kugel ins Aus gehen lassen und man erhält ein schwierigeres Freispiel.

Michael Vorwieger

```
100  PZ=0
110  SGE=6
120  CALL CLEAR
130  REM  ZEICHENDEFINIEREN
140  SGE=SGE+1
150  CALL CHAR(150,"FFFFFFFFF
FFFFFFF")
160  CALL CHAR(152,"3333CCCC3
333CCCC")
170  CALL CHAR(140,"FFFFFFFF"
)
180  CALL CHAR(141,"FFFFFFFF"
)
190  CALL CHAR(130,"3C7EFFFFF
FFF7E3C")
200  CALL HCHAR(1,7,152,21)
210  CALL VCHAR(1,27,152,22)
220  CALL VCHAR(1,7,152,22)
230  FOR RT=3 TO SGE
240  CALL HCHAR(RT,9,150,17)
250  NEXT RT
260  WK=23
270  SK=17
280  WA=22
290  SE=15
300  CALL HCHAR(WK,SK+1,141)
310  CALL HCHAR(WK,SK,140)
320  CALL SCREEN(8)
330  CALL COLOR(15,5,1)
340  CALL COLOR(13,7,1)
350  CALL COLOR(16,7,2)
360  CALL HCHAR(1,4,SGE+42)
400  CALL SOUND(50,-1,1)
410  WA=WA-1
420  SE=SE+1
430  A=1
440  CALL HCHAR(WA+1,SE-1,32)
450  CALL HCHAR(WA,SE,130)
460  CALL HCHAR(WA+1,SE-1,32)
470  IF SE=26 THEN 600
480  IF WA=2 THEN 1030
490  CALL GCHAR(WA-1,SE,TM)
500  IF TM=150 THEN 1000
510  GOTO 1200
600  CALL SOUND(50,-2,1)
610  WA=WA-1
620  SE=SE-1
630  A=2
640  CALL HCHAR(WA+1,SE+1,32)
650  CALL HCHAR(WA,SE,130)
660  CALL HCHAR(WA+1,SE+1,32)
670  IF SE=8 THEN 400
680  IF WA=2 THEN 830
690  CALL GCHAR(WA-1,SE,TC)
700  IF TC=150 THEN 800
710  GOTO 1200
800  CALL HCHAR(WA-1,SE,32)
810  CALL SOUND(50,-7,0)
820  REM
830  CALL SOUND(50,-3,1)
840  A=3
850  WA=WA+1
860  SE=SE-1
870  CALL HCHAR(WA-1,SE+1,32)
880  CALL HCHAR(WA,SE,130)
890  CALL HCHAR(WA-1,SE+1,32)
900  IF SE=8 THEN 1030
910  IF WA>=23 THEN 1500
920  CALL GCHAR(WA+1,SE,R6)
930  IF R6=140 THEN 600
940  IF R6=141 THEN 400
950  GOTO 1200
1000 CALL HCHAR(WA-1,SE,32)
1010 CALL SOUND(50,-7,0)
1020 REM
1030 CALL SOUND(50,-3,1)
1040 WA=WA+1
1050 SE=SE+1
1060 A=4
1070 CALL HCHAR(WA-1,SE-1,32
)
1080 CALL HCHAR(WA,SE,130)
1090 CALL HCHAR(WA-1,SE-1,32
)
1100 IF SE=26 THEN 830
1110 IF WA>=23 THEN 1500
1120 CALL GCHAR(WA+1,SE,RM)
1130 IF RM=141 THEN 400
1140 IF RM=140 THEN 600
1151 1131-1160
1140 IF RM=140 THEN 600
1150 GOTO 1200
1200 CALL KEY(0,KE,ST)
1210 IF ST=0 THEN 1240
1220 IF KE=44 THEN 1250
1230 IF KE=46 THEN 1310
1240 ON A GOTO 410,610,840,1
040
1250 IF SK=8 THEN 1240
1260 CALL HCHAR(WK,SK-1,140)
1270 CALL HCHAR(WK,SK,141)
1280 CALL HCHAR(WK,SK+1,32)
1290 SK=SK-1
1300 ON A GOTO 410,610,840,1
040
1310 IF SK=26 THEN 1240
1320 CALL HCHAR(WK,SK,140)
1330 CALL HCHAR(WK,SK+1,141)
1340 CALL HCHAR(WK,SK-1,32)
1350 SK=SK+1
1360 ON A GOTO 410,610,840,1
040
1500 FOR WQ=3 TO SGE
1510 FOR SQ=9 TO 25
1520 CALL GCHAR(WQ,SQ,TQ)
1530 IF TQ=150 THEN 1550
1540 PZ=PZ+5
1550 NEXT SQ
1560 NEXT WQ
1570 IF PZ=425 THEN 120
1580 IF PZ=935 THEN 120
1590 IF PZ=1530 THEN 120
1595 IF PZ=2210 THEN 120
1600 DRUCK$="PUNKTZAHL = "&S
TR$(PZ)
1610 VTAB=8
1620 HTAB=9
1630 GOSUB 1760
1640 IF HS>PZ THEN 1670
1650 HS=PZ
1660 GOSUB 1940
1670 DRUCK$="HIGHSCORE = "&S
TR$(HS)
1680 GOSUB 1760
1690 DRUCK$="NOCH EIN SPIEL?
"
1700 GOSUB 1760
1702 DRUCK$="<J> <N> "
1704 GOSUB 1760
1710 CALL KEY(0,KL,SL)
1720 IF SL=0 THEN 1710
1730 IF KL=74 THEN 100
1740 IF KL<>78 THEN 1710
1750 END
1760 REM  PRINTROTINE
1770 FOR DRI=1 TO LEN(DRUCK$
)
1780 CODE=ASC(SEG$(DRUCK$,DR
I,1))
1790 GOSUB 1830
1800 NEXT DRI
1810 GOSUB 1880
1820 RETURN
1830 REM  COUT
1840 CALL HCHAR(VTAB,HTAB,CO
DE)
1850 HTAB=HTAB+1
1860 IF HTAB<32 THEN 1930
1870 REM  CROUT
1880 VTAB=VTAB+1
1890 HTAB=9
1900 IF VTAB<25 THEN 1930
1910 VTAB=24
1920 PRINT
1930 RETURN
1940 FOR SI=1 TO 5
1950 FOR SJ=1100 TO 880 STEP
-110
1960 CALL SOUND(100,SJ,1)
1970 NEXT SJ
1980 NEXT SI
1990 RETURN
2000 REM  HA,HA
```

mir natürlich vorstellen, daß einige Softwarenanbieter böse reagieren. Wenn das so weitergehen sollte, so wird nach meiner Schätzung diese Ausgabe zu einem dauerhaften Renner werden. Es wäre allerdings zu wünschen, daß die Listen für den ZX größer und sauberer heraus kämen. Ein Normalpapierdrucker könnte sich möglicherweise doch lohnen.
Nun noch zu Ihrem Softwareservice. Ich frage mich, ob Sie das auf die Dauer verkraften können. Die Preise sind mit dem Ausdruck günstig nicht beschrieben. Ich habe noch keinen Gebrauch von Ihrem Angebot gemacht. Aber ich hoffe doch, daß jedes Programm wenigstens zwei mal geladen worden ist. Drop Outs kommen manchmal schon vor. Eine zweite Aufnahme (nicht paraellel) bietet nahezu 100%ige Datensicherheit.
Zu Ihrer Schrift kann ich sagen, daß diese nicht so schlecht ist, wie sie offensichtlich von einigen Lesern dargestellt wird. Sie haben viel Geld und Zeit investiert, um diese Schrift zu entwerfen, und so ist es klar, daß Sie mindestens noch einige Zeit an dieser Schriftform festhalten wollen.

R. Schuff — Hochspeyer

Kürzlich entdeckte ich Ihre neue Zeitschrift und habe sie mal probeweise gekauft. Überrascht war ich von den – wie ich meine – guten und interessanten Programmen für den ZX-81.
Die Mutmaßung des Autors des Programms 'LIVE', daß es Perioden mit einer Länge von mehr als 6 vielleicht nicht geben könne, ist meines Erachtens von den 'Spiel'-regeln her unbegründet und auch nicht wahr: z. B. stellt die Konfiguration 0102030405111216242526 eine 14-er Periode dar. Setzt man in eine andere Ecke noch z. B. die 3-er Periode f1f2f3e0e5d0d5c0c5a1a2a3, so erhält man insgesamt eine Periode von 42 (welches das kleinste gemeinsame Vielfach ist). Entsprechend kann man noch höhere Persioden finden, wenn man nur Perioden der Länge n findet, wobei n kein Teiler von 42 sein darf, aber wie der Autor schon sagt, ist das nicht leicht.
Ich glaube, daß Ihre Zeitung großen Zuspruch finden wird, wenn Sie weiterhin solche Programme bringen werden und vielleicht ein besseres Schriftbild, gerade bei den Listings der Maschinenprogramme, die fast ohne Redundanz sind, einführen.

Richard Bauer — Bonn

Als erstes möchte ich Ihnen gratulieren, die Zeitung „Homecomputer" ist Ihnen gut gelungen.
Nun ein Verbesserungsvorschlag für das ZX-Programm LIFE. Da nach dem Setzen eines Steines durch die Taste P und der danach weiteren Steuerung des Cursor der gesetzte Stein wieder verschwindet, würde ich das Programm um folgende Zeile ergänzen.

```
2255 IF S=$ THEN PRINT AT Z,K; CHR$ 18Ø
```

Hierdurch wird nach dem Weitersteuern des Cursor der Stein erneut gesetzt (sofern ein Stein gesetzt wurde).
Zu dem Programm SURVIVE und LABYRINTH für den ZX. Haben sich hier Fehler im Maschinenlisting eingeschlichen, mit Fehlermeldung. Zum Teil sind auch die Zeichen des Maschinenprogrammes nicht einwandfrei zu definieren (am Rand), besonders die 8 und das B.
Bezüglich dem Programm Spukhaus für den VC-20 würde ich vorschlagen, solche Listing im Graphik-Modus auszulisten, wie bei Maze-Challenger, da dann die graphischen Zeichen (ab Zeile-Nr. 5ØØØØ) besser zu unterscheiden sind, was die Eingabe wesentlich erleichtert.
Für die Zukunft Ihrer Zeitschrift wünsche ich Ihnen viel Erfolg und verbleibe

Wolfgang Wieber — Mainz

Ich möchte Ihnen mitteilen, das das Programm für den TI99 aus Heft 4 einen kleinen Fehler hat: und zwar muß die Zeile 855 PZ=Ø eingefügt werden, weil sonst bei einem neuen Spiel der alte Wert von PZ genommen wird und die neuen Punkte dazuaddiert werden, so daß man kein Ergebnis für das letzte Spiel erhält.
Ich habe das Programm wie folgt geändert um es optisch und akustisch interessanter zu machen:

```
120 CALL CHAR(96,"00183C5ABD995A18")
130 CALL CHAR(152,"187871CFE3(1")
140 CALL CHAR(153,"3F1DAFF97FF4(23E")
370 IF K=49 THEN 425          (Anm.: Taste 49 + 47
380 IF K=47 THEN 455           sind handl. als 81 + 65)
425 CALL SOUND(-50,-5,2)
455 CALL SOUND(-50,-7,2)
800 CALL SOUND(-10,-5,2)
801 CALL SOUND(-10,-6,2)
802 CALL SOUND(-10,-7,2)
803 CALL SOUND(-10,-6,2)
804 CALL SOUND(-10,-5,2)
805 Druck$="Punktzahl = "&STR$(PZ)
```

(Anm.: Zeile 805 ist im Originalprogramm die Zeile 800.)

P.S. Homecomputer ist „klasse"!

Dirk Kessina — Essen

Aus Ihrer Zeitschrift 4/80 habe ich das Programm „Labyrinth für den VC 20" entnommen. Leider ist mir aufgefallen, daß manchmal die senkrechten Striche der Trennwände fehlten. Darum schlage ich vor die Zeile 20 so zu schreiben:

```
20 IFY=32ANDZ=32THENFORA=TC-U+LLTOBC-
V-LISTEPLL:POKER,W3:POKEA+VP,CO:NEXT
```

Durch dieses Einfügen wird die 3-D-Wirkung wesentlich verbessert, aber noch besser ist es den Wert W3 in Zeile 955 von 102 auf 160 zu erhöhen.
Vielleicht ist es der gleiche Erfolg auch beim VC 64, da das Programm fast gleich ist.

Dirk Büchner — Schwetzingen

Steckerspiel

Die Regeln sind aus dem Listing ersichtlich. Wer nicht so viel abschreiben möchte, kann die Zeilen 210 bis 340 und 1170 ff (Spielanleitung), sowie alle REMs ersatzlos streichen, füge allerdings die Zeile

210 CALL CLEAR hinzu.
Achtung: Die Unlaute ÄÖÜ sind im Programm definiert, deshalb bitte die kleinen Buchstaben (a,o,u) wie im Listing schreiben.

```
100 !STECKERSPIEL
110 !DEFINIEREN VON ZEICHEN
   SAETZE,DIMENSIONEN,FARBE
   UMLAUTE
120 DEF BL(X)=INT(X/3*5)
130 OPTION BASE 1 :: DIM A(8
,8),U(4),H(2),ZG$(64),S$(15)
,SPI$(2)
140 SPI$(1)="111111111111111
111000011110000111100001111 0
00011111111111111111"
150 S$(1)="ZUG        1234567
801" :: S$(2)="VON
 ! LETZTER ZUG12" :: S$(3)="
NACH14" :: S$(4)="BEI FEHLEI
NGABE       (FCTN3)20"
160 S$(5)="ZUGRuCKNAHME
    (FCTN9)24" :: S$(6)="WIE
DERHOLUNG          (FCTN8)23"
 :: S$(7)="AUSWERTUNG
    (FCTN7)22"
170 S$(8)="FEHLER      FEHLER
     FEHLER" :: S$(9)="
   AUSWERTUNG" :: S$(10)="VO
N  STIFTEN WURDEN" :: S$(11)
=" STIFTE ENTFERNT"
180 S$(12)="DIES SIND  PROZE
NT." :: S$(13)="NOTE " :: S$
(14)="NEUES SPIEL          (
ENTER)" :: S$(15)="RuCKNAHME
"
190 CALL CHAR(97,"0044102844
7C4444",111,"004438444444443
8",117,"0044004444444438",12
7,"000000181800000000245A666
65A24")
200 CALL COLOR(12,13,16,13,1
4,16,14,9,9):: CALL SCREEN(1
2)
210 DISPLAY AT(10,8)ERASE AL
L:"STECKERSPIEL":"        ===
=========" :: DISPLAY AT(24,
1):"ANLEITUNG ERWuNSCHT?   (
J/N)"
220 CALL KEY(3,K,ST):: IF K<
>74 AND K<>78 THEN 220
230 IF K=78 THEN CALL CLEAR
:: GOTO 350
240 !SPIELANLEITUNG
250 DISPLAY AT(1,1)ERASE ALL
:"AUF EINEM 8 X 8-FELDER-BRE
TTSIND STIFTE VERTEILT.
 (STANDARTSPIEL AUF DEN ZWEI
 aUSSEREN REIHEN)"
260 DISPLAY AT(6,1):"DURCH u
BERSPRINGEN MuSSEN   SOVIEL
STIFTE WIE MoGLICH   ENTFERN
T WERDEN."
270 DISPLAY AT(10,1):"EIN ZU
G BESTEHT DARIN DAS   EIN ST
IFT DIAGONAL(schra9)  IN EIN
 FREIES FELD GEZOGEN  WIRD."
280 DISPLAY AT(15,1):"ALLE A
NDEREN VERSUCHE HABEN FEHLER
MELDUNGEN ZUR FOLGE."
290 DISPLAY AT(18,1):"NACH E
INER FEHLERMELDUNG:": :"1.uB
ERPRuFE LETZTE EINGABE":"2.G
EBE NEU EIN." :: GOSUB 1170
300 DISPLAY AT(3,1):"WENN DI
E EINGABE GEaNDERT   WERDEN
SOLL KANN DIES BEI   DER EIN
GABE NACH DRuCKEN    VON (FC
TN3) GESCHEHEN."
310 DISPLAY AT(9,1):"GEMACHT
E ZuGE KoNNEN DURCH  DRuCKEN
 VON (FCTN9)  ZURuCK-GENOMME
N WERDEN."
320 DISPLAY AT(14,1):"ZUR WI
EDERHOLUNG DES BISHE- RIGEN
SPIELES DRuCKE (FCTN8)": : :
"ZUR AUSWERTUNG DRuCKE(FCTN7
)" :: GOSUB 1170
330 DISPLAY AT(5,1):"DIE NOT
EN BEI DER AUSWERTUNG": :"
     1=SEHR GUT": :"
   BIS": :"    12=HUNDSMISER
ABEL" :: GOSUB 1170
340 DISPLAY AT(5,1):"SIE KoN
NEN JETZT WaHLEN ZWI-SCHEN S
TANDARTSPIEL UND EI- GENEN S
PIELFELDAUFBAU."
350 DISPLAY AT(12,1):"STANDA
RTSPIEL        DRuCKE 1": :"E
IGENE AUFSTELLUNG  DRuCKE 2"
360 CALL KEY(3,JJ,ST):: IF J
J<49 OR JJ>50 THEN 360
370 JJ=JJ-48 :: IF JJ=1 THEN
 390 ELSE GOSUB 1190
380 !SPIELFELDAUFBAU
390 CALL CLEAR
400 A$=SPI$(JJ):: STE,RR,D=0
 :: FOR Z=1 TO 8 :: CALL HCH
AR(1+Z,12,64+Z)
410 FOR S=1 TO 8
420 D=D+1 :: A(Z,S)=VAL(SEG$
(A$,D,1)):: STE=STE+A(Z,S)::
 CALL HCHAR(1+Z,12+S,127+A(Z
,S))
430 NEXT S :: NEXT Z
440 IF SP1=1 THEN 1130
450 FOR T=1 TO 7 :: DISPLAY
AT(VAL(SEG$(S$(T),LEN(S$(T))
-1,2)),1):SEG$(S$(T),1,LEN(S
$(T))-2):: NEXT T
460 CALL HCHAR(13,3,45,11)::
 CALL HCHAR(13,17,33):: CALL
 HCHAR(13,24,47)
470 CALL HCHAR(12,7,32,9)::
CALL HCHAR(14,7,32,9)
480 !ZUGEINGABE
490 OO,P=0
500 FOR EN1=1 TO 2 :: Y=0
510 FOR EN2=1 TO 2 :: P=P+1
520 CALL HCHAR(11+BL(EN1),13
,60,2)
530 CALL KEY(3,K,ST):: IF ST
<1 THEN 530
540 CALL HCHAR(11+BL(EN1),8+
BL(EN2),K)
550 IF K=1 AND RR>0 THEN 930
560 IF K=7 THEN 470
570 IF K=15 AND RR>0 THEN GO
SUB 1040
580 IF K=6 AND RR>0 THEN GOS
UB 1120
590 KK=K-48-16*ABS(EN2-2)::
IF KK<1 OR KK>8 THEN 530
600 U(P),H(EN2)=KK :: CALL H
CHAR(11+BL(EN1),13,32,2):: N
EXT EN2
610 IF Y<4 THEN ON EN1 GOSUB
 850,870 :: GOTO 610
620 IF OO=1 THEN 470
630 !UEBERPRUEFEN AUF
     GUELTIGKEIT
640 IF ABS(SGN(EN1-2))<>A(H(
1),H(2))THEN GOSUB 900 :: GO
TO 470 ELSE 650
650 NEXT EN1
660 IF ABS(U(1)-U(3))<>2 OR
ABS(U(2)-U(4))<>2 THEN GOSUB
 900 :: GOTO 470 ELSE 670
670 GOSUB 690 :: GOSUB 800 :
: GOTO 470
680 !ZUGAUSFUEHRUNG
690 S1=U(1)-(U(1)-U(3))/2 ::
 R1=U(2)-(U(2)-U(4))/2
700 IF A(S1,R1)=1 OR K=15 OR
 K=6 THEN 720
710 GOSUB 900 :: GOTO 470
720 A(U(1),U(2))=ABS(A(U(1),
U(2))-1)
730 CALL HCHAR(1+U(1),12+U(2
),127+A(U(1),U(2)))
740 A(S1,R1)=ABS(A(S1,R1)-1)
750 CALL HCHAR(1+S1,12+R1,12
7+A(S1,R1))
760 A(U(3),U(4))=ABS(A(U(3),
U(4))-1)
770 CALL HCHAR(1+U(3),12+U(4
),127+A(U(3),U(4)))
780 RETURN
790 !ZUGKOORDINATEN
800 RR=RR+1 :: ZG$(RR)="" ::
 FOR T=1 TO 4 :: ZG$(RR)=ZG$
(RR)&STR$(U(T)):: NEXT T
810 !ZUG AUF BILDSCHIRM
     SCHREIBEN
820 A$="" :: FOR T=1 TO 3 ST
EP 2 :: FOR I=0 TO 1 :: A$=A
$&CHR$(ASC(SEG$(ZG$(RR),T+I,
1))+16*ABS(I-1))&" " :: NEXT
 I :: A$=A$&" /  " :: NEXT T
830 DISPLAY AT(3,1)SIZE(3):R
R :: DISPLAY AT(13,17)SIZE(1
1):A$ :: RETURN
840 !BLINKEN
850 Y=Y+1 :: CALL GCHAR(1+U(
```

```
1),12+U(2),X1):: CALL HCHAR(
1+U(1),12+U(2),32)
850 FOR AA=1 TO 20 :: NEXT A
A :: CALL HCHAR(1+U(1),12+U(
2),X1):: RETURN
870 CALL GCHAR(1+U(3),12+U(4
),X2):: CALL HCHAR(1+U(3),12
+U(4),32)
880 FOR AA=1 TO 20 :: NEXT A
A :: CALL HCHAR(1+U(3),12+U(
4),X2):: GOSUB 850 :: RETURN
890 !FEHLERROUTINE
900 FOR F=1 TO 3 :: CALL HCH
AR(17,1,136,32):: DISPLAY AT
(16,1):S$(8):: CALL SOUND(10
0,110,0):: FOR T=1 TO 150 ::
 NEXT T
910 CALL HCHAR(16,1,32,64)::
 FOR T=1 TO 150 :: NEXT T ::
 NEXT F :: RETURN
920 !AUSWERTUNG
930 CALL HCHAR(11,1,32,448)
940 DISPLAY AT(11,1):S$(9):
:SEG$(S$(10),1,4)&STR$(STE)&
SEG$(S$(10),5,15)
950 DISPLAY AT(15,1):RR;S$(1
1): :SEG$(S$(12),1,10)&STR$(
INT(100*RR/(STE-1)))&SEG$(S$
(12),11,9): :S$(13):INT((STE
-RR)/4+.75)
960 !
970 CALL HCHAR(22,1,32,96)::
 DISPLAY AT(23,1):S$(14):: D
ISPLAY AT(24,1)SIZE(28):S$(6
)
980 !ABFRAGE OB NEUES SPIEL
990 CALL KEY(3,K,ST):: IF ST
=0 THEN 990 ELSE IF K=6 THEN
 1010
1000 IF K<>13 THEN DISPLAY A
T(15,1):"NEUES SPIEL NUR MIT
 <RUN>" :: END ELSE CALL CLE
AR :: GOTO 350
1010 CALL HCHAR(12,1,32,416)
1020 GOSUB 1120 :: GOTO 970
1030 !ROUTINE ZUR ZUGRUECKN.
1040 DISPLAY AT(15,17):S$(15
):: CALL HCHAR(16,19,136,9)
1050 RESTORE 1060
1060 DATA 3,4,1,2
1070 FOR ZU=1 TO 4 :: READ Z
 :: U(ZU)=VAL(SEG$(ZG$(RR),Z
,1)):: NEXT ZU :: Y=0
1080 IF Y<4 THEN GOSUB 870 :
: GOTO 1080
1090 GOSUB 690 :: ZG$(RR)=""
 :: RR=RR-1 :: CALL HCHAR(3,
2,32,5):: CALL HCHAR(13,19,3
2,3):: CALL HCHAR(13,27,32,3
)
1100 IF RR<>0 THEN GOSUB 820
 ELSE Y=0 :: CALL HCHAR(15,1
9,32,11):: CALL HCHAR(16,19,
32,11):: RETURN
1110 !SPIEL NOCHEINMAL SEHEN
1120 DISPLAY AT(15,17):"WIED
ERHOLUNG" :: CALL HCHAR(16,1
9,136,12):: SP1=1 :: OX=RR :
: GOTO 400
1130 FOR AA=1 TO 4 :: U(AA)=
VAL(SEG$(ZG$(RR+1),AA,1)):: 
NEXT AA :: Y=0
1140 IF Y<4 THEN GOSUB 870 :
: GOTO 1140
1150 GOSUB 690 :: GOSUB 800
:: IF RR<>OX THEN 1130
1160 CALL HCHAR(15,1,32,64):
: SP1=0 :: RETURN
1170 DISPLAY AT(24,1):"ENTER
 DRuCKEN" :: ACCEPT AT(24,14
):Z$ :: CALL CLEAR :: RETURN
1180 !EIG. SPIELFELDAUFBAU
1190 CALL CLEAR :: SPI$(2)="
" :: FOR T=1 TO 8 :: CALL HC
HAR(1+T,13,127,8):: NEXT T
1200 DISPLAY AT(13,1):"STEUE
R DAS ZEICHEN # MIT DENTASTE
N  E,X,S,D": :"STIFT
 DRuCKE 1": :"KEIN STIFT
DRuCKE 0"
1210 DISPLAY AT(21,1):"FuR A
NFANG    DRuCKE A"
1220 CALL SPRITE(#1,35,2,9,9
8)
1230 CALL KEY(1,K,ST):: IF S
T=0 THEN CALL MOTION(#1,0,0)
:: GOTO 1230
1240 IF K=1 THEN 1320 ELSE I
F K=19 OR K=18 THEN CALL POS
ITION(#1,Y,X):: Y=Y/8+1 :: X
=X/8+1 :: CALL HCHAR(Y,X,127
+K-18)
1250 IF K=4 OR K>5 THEN 1230
1260 ON K+1 GOSUB 1280,1340,
1290,1300,1340,1310
1270 CALL MOTION(#1,5*YY,5*X
X):: CALL POSITION(#1,Y,X):
 IF Y<6 OR Y>70 OR X<96 OR X
>158 THEN CALL DELSPRITE(#1)
:: GOTO 1220 ELSE 1230
1280 YY=1 :: XX=0 :: RETURN
1290 XX=-1 :: YY=0 :: RETURN
1300 XX=1 :: YY=0 :: RETURN
1310 YY=-1 :: XX=0 :: RETURN
1320 DISPLAY AT(24,1):"BITTE
 WARTEN" :: CALL DELSPRITE(#
1)
1330 FOR Z=1 TO 8 :: FOR S=1
 TO 8 :: CALL GCHAR(1+Z,12+S
,X):: SPI$(2)=SPI$(2)&STR$(X
-127):: NEXT S :: NEXT Z
1340 RETURN
```

Mampfmann für ZX Spectrum

Ein äußerst lustiges Spiel für den Spectrum, bei dem der Spieler beim Äpfelklauen auf gefährliche kleine Monster stößt, die einem die Freude an Nachbars Äpfeln mächtig vergällen. Auch der Zaun ist äußerst scharf geladen und sollte auf keinen Fall berührt werden.

```
          MAPFMANN
       by Gerd Proschek

       © Gerd Proschek
          Hannover


   5 PRINT BRIGHT 1; FLASH 1; IN
K 2;"*********  MAMPFMANN  *****
****"; BRIGHT 0;AT 5,6; FLASH 0;
 INK 0;"by Gerd Proschek


          © Gerd Proschek
             Hannover"
   6 GO TO 9000
   7 CLS
 100 REM Einleitung
 110 CLS
 120 PRINT INK 2; FLASH 1; BRIGH
T 1;"*********  MAMPFMANN  *****
****"; FLASH 0; BRIGHT 0
 130 PRINT INK 0,,,,"Sie sind mi
t ihrem Mampfer (";INK 2; FLASH
 1;"▩"; INK 0; FLASH 0;")  in Mo
nster's Garten geschlichen und v
ersuchen dort die Äpfel(";INK 4
;"●"; INK 0;") zu stehlen."
 140 PRINT "Leider stehen dort d
ie kleinen  Kinder des Monsters(
"; INK 3;"▚"; INK 0;") und auch
der Monster-Vater (";INK 3;"▚";
 INK 0;") läuft hin und her, und
 Sie bewachen im     Garten die A
pfel !"
 142 PRINT "Diese Monster sind s
ehr agressivund gefärlich ! Vers
uchen Sie   ihnen also auszuweic
hen ! Auch  den Zaun dürfen Sie
nicht       berühren, denn er is
t sehr      scharf !"
 144 PRINT AT 21,0;"Drücken Sie
eine Taste"
 145 PAUSE 10: PAUSE 5000
 146 CLS
 147 PRINT INK 2; BRIGHT 1; FLAS
H 1;"*********  MAMPFMANN  *****
****"; FLASH 0; BRIGHT 0
 151 PRINT ,,,,"Falls Sie doch e
inmal ein      Monster oder den
 Zaun berühren  sollten, kostet
sie das eines   Ihrer 5 'Leben'.
 Sie bekommen   jedoch alle 75 P
unkte ein neues 'Leben' wieder d
azu."
 152 PRINT : PRINT "Wenn Sie in
einem Garten alle   Äpfel gemamp
ft haben,können Sie durch den Au
sgang(rechts)in den nächsten Gar
ten laufen."
 153 PRINT AT 21,0;"Drücken Sie
eine Taste"
 160 PAUSE 5000
 200 CLS
 205 PRINT AT 0,14;"Oben";AT 9,1
0;"Geben Sie","Links      die Tas
ten      Rechts";TAB 13;"ein";AT
20,14;"Unten": PAUSE 10
 206 PRINT AT 2,16; FLASH 1; BRI
GHT 1; INK 2;"▩"
 207 LET r$=INKEY$: IF r$="" THE
N GO TO 207
 208 PRINT AT 2,16; FLASH 0;r$
 209 PRINT AT 10,6; FLASH 1; BRI
GHT 1;"▩"
 210 LET d$=INKEY$: IF d$="" OR
d$=r$ THEN GO TO 210
 211 PRINT AT 10,6; FLASH 0;d$
 212 PRINT AT 10,23; FLASH 1; BR
IGHT 1;"▩"
 213 LET g$=INKEY$: IF g$="" OR
g$=d$ THEN GO TO 213
 214 PRINT AT 10,23; FLASH 0;g$
 215 PRINT AT 18,16; FLASH 1; BR
IGHT 1;"▩"
 216 LET c$=INKEY$: IF g$=c$ OR
c$="" THEN GO TO 216
 217 PRINT AT 18,16; FLASH 0;c$:
 PAUSE 10
 218 PRINT AT 21,8;"Richtig so ?
(J/N)"
 219 IF INKEY$="j" OR INKEY$="J"
 THEN GO TO 230
 220 IF INKEY$="n" OR INKEY$="N"
 THEN GO TO 200
 225 GO TO 219
 230 CLS
 235 PRINT INK 2; FLASH 1; BRIGH
T 1;"*********  MAMPFMANN  *****
****"; FLASH 0; BRIGHT 0
 240 PRINT AT 5,0; INK 0;"Wählen
 Sie die Schwierigkeits-  stufe
!    (0-9) "  ,,,,,"      Am schnell
sten ist 0 !"
 250 LET t$=INKEY$: IF t$<"0" OR
 t$>"9" THEN GO TO 250
 260 LET t=VAL t$*1.5
 270 PRINT ,,"Sie wählten die
                Spielgeschwindigke
it ";t$
 280 PRINT ,,,,"Soll der Monster
-Vater                 herumlaufen?
                          (J/N)"
```

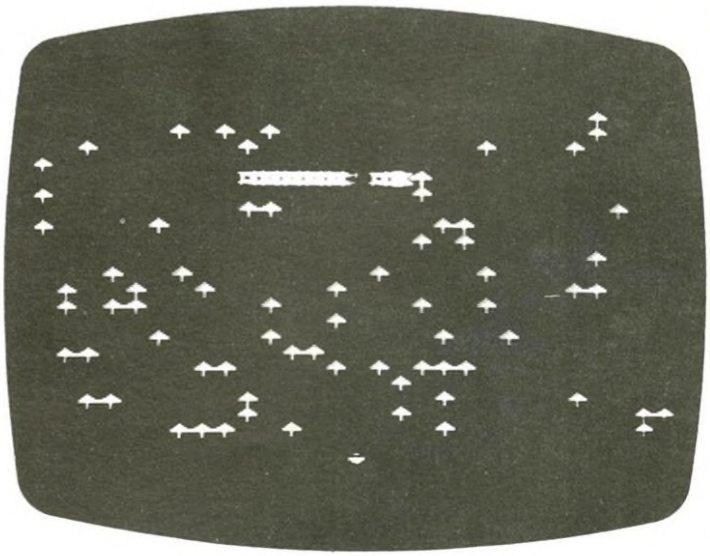
Der Pilzwurm

Der erbitterte Kampf gegen den immer näher rückenden Pilzwurm ist schier aussichtslos.

Er ist nur kurzzeitig zu verletzen, regeneriert aber immer wieder. Je öfter er an ein Hindernis stößt, desto schneller kommt er näher. Also müssen die Pilze weg, bevor er anstößt.
Schleudern Sie dem Pilzwurm und den Pilzen Ihren Giftzahn entgegen.
Wertetabelle:

Pilz	- 100 Punkte
Wurmkörper	- 200 Punkte
Wurmkopf	- 500 Punkte

Gesteuert wird die Schlange mit den Cursorsteuertasten S und D und der Semikolontaste.

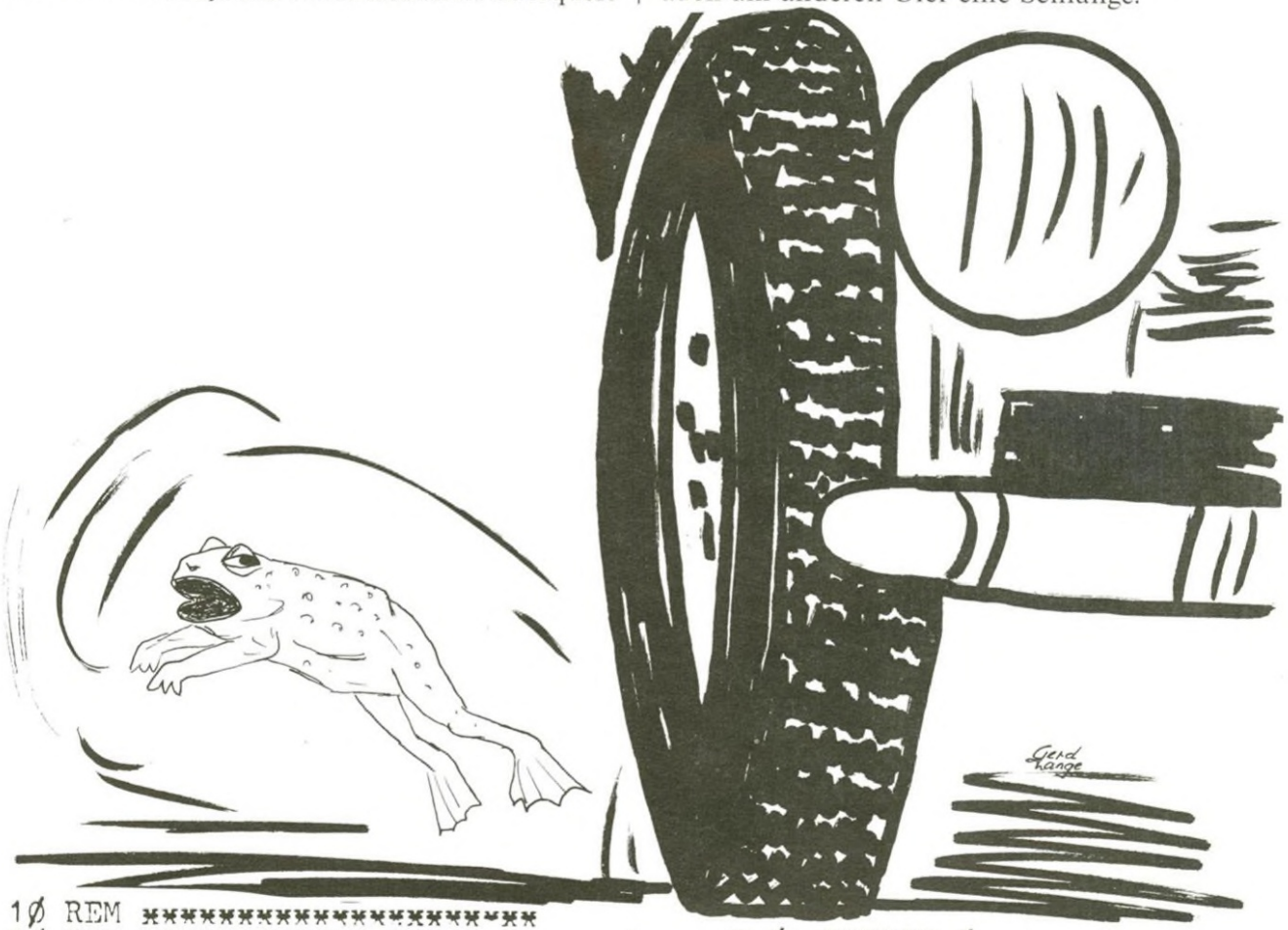


```
100  CALL  CLEAR
110  PRINT "*****************
************
                **              PILZWUR
M               *"
120  PRINT      "*
                ****************
************":::" VON JUERGEN
 HOPPSTAEDTER":::
130  FOR  I=1  TO  1000
140  NEXT  I
150  CALL  CLEAR
160  RANDOMIZE
170  CALL  CHAR(128,"103854FEF
E7C3838")
180  CALL  CHAR(136,"183C7EFFF
F181818")
190  CALL  CHAR(144,"78FD767C7
C76FD78")
200  CALL  CHAR(145,"42FFFFFFD
B7E2442")
210  CALL  CHAR(146,"1EBF6E3E3
E6EBF1E")
220  CALL  CHAR(147,"42FF7E666
67EFF42")
230  CALL  CHAR(152,"001010101
0")
240  CALL  SCREEN(2)
250  CALL  COLOR(13,7,1)
260  CALL  COLOR(14,13,1)
270  CALL  COLOR(15,5,1)
280  CALL  COLOR(16,12,1)
290  FOR  I=1  TO  90
300  H=INT(21*RND)+2
310  V=INT(30*RND)+2
320  CALL  SOUND(50,440,5)
330  CALL  HCHAR(H,V,136)
340  NEXT  I
350  CALL  HCHAR(24,16,128)
360  CALL  SOUND(300,262,5)
370  CALL  SOUND(900,349,5)
380  CALL  SOUND(100,392,5)
390  CALL  SOUND(100,349,5)
400  CALL  SOUND(100,392,5)
410  CALL  SOUND(300,440,5)
420  CALL  SOUND(300,440,5)
430  CALL  SOUND(300,440,5)
440  CALL  SOUND(100,440,5)
450  CALL  SOUND(100,466,5)
460  CALL  SOUND(100,392,5)
470  CALL  SOUND(450,440,5)
480  CALL  SOUND(150,392,5)
490  CALL  SOUND(300,349,5)
500  CALL  SOUND(100,-8,5)
510  CALL  HCHAR(1,32,146)
520  CALL  SOUND(100,-8,5)
530  CALL  HCHAR(1,31,146)
540  CALL  HCHAR(1,32,147)
550  CALL  SOUND(100,-8,5)
560  CALL  HCHAR(1,30,146)
570  CALL  HCHAR(1,31,147)
580  CALL  SOUND(100,-8,5)
590  CALL  HCHAR(1,29,146)
600  CALL  HCHAR(1,30,147)
610  CALL  SOUND(100,-8,5)
620  CALL  HCHAR(1,28,146)
630  CALL  HCHAR(1,29,147)
640  CALL  SOUND(100,-8,5)
650  CALL  HCHAR(1,27,146)
660  CALL  HCHAR(1,28,147)
670  CALL  SOUND(100,-8,5)
680  CALL  HCHAR(1,26,146)
690  CALL  HCHAR(1,27,147)
700  CALL  SOUND(100,-8,5)
710  CALL  HCHAR(1,25,146)
720  CALL  HCHAR(1,26,147)
730  D=-1
740  P=0
750  W=16
760  X=1
770  Y=25
780  CALL  KEY(0,KEY,STATUS)
790  V=24
800  IF  KEY=59  THEN  1510
810  IF  KEY=83  THEN  1460
820  IF  KEY=115  THEN  1460
830  IF  KEY=68  THEN  1410
840  IF  KEY=100  THEN  1410
850  Y=Y+D
860  IF  Y=1  THEN  960
870  IF  Y=32  THEN  960
880  CALL  GCHAR(X,Y,Z)
890  IF  X=24  THEN  1810
900  IF  Z=136  THEN  960
910  CALL  SOUND(100,-8,5)
920  CALL  HCHAR(X,Y,145-D)
930  CALL  HCHAR(X,Y-D,147)
940  CALL  HCHAR(X,Y-8*D,32)
950  GOTO  780
960  CALL  SOUND(100,-3,5)
970  D=-D
980  X=X+1
990  IF  X=24  THEN  1810
1000  Y=Y+D
1010  CALL  SOUND(100,-8,5)
1020  CALL  HCHAR(X,Y,145-D)
1030  CALL  HCHAR(X-1,Y,147)
1040  CALL  HCHAR(X-1,Y+7*D,32
1050  Y=Y+D
1060  CALL  SOUND(100,-8,5)
1070  CALL  HCHAR(X,Y,145-D)
1080  CALL  HCHAR(X,Y-D,147)
1090  CALL  HCHAR(X-1,Y+5*D,32
1100  Y=Y+D
1110  CALL  SOUND(100,-8,5)
1120  CALL  HCHAR(X,Y,145-D)
1130  CALL  HCHAR(X,Y-D,147)
1140  CALL  HCHAR(X-1,Y+3*D,32
)
1150  Y=Y+D
1160  CALL  SOUND(100,-8,5)
1170  CALL  HCHAR(X,Y,145-D)
1180  CALL  HCHAR(X,Y-D,147)
1190  CALL  HCHAR(X-1,Y+D,32)
1200  Y=Y+D
1210  CALL  SOUND(100,-8,5)
1220  CALL  HCHAR(X,Y,145-D)
1230  CALL  HCHAR(X,Y-D,147)
1240  CALL  HCHAR(X-1,Y-D,32)
1250  Y=Y+D
1260  CALL  SOUND(100,-8,5)
1270  CALL  HCHAR(X,Y,145-D)
1280  CALL  HCHAR(X,Y-D,147)
1290  CALL  HCHAR(X-1,Y-3*D,32
)
1300  Y=Y+D
1310  CALL  SOUND(100,-8,5)
1320  CALL  HCHAR(X,Y,145-D)
1330  CALL  HCHAR(X,Y-D,147)
1340  CALL  HCHAR(X-1,Y-5*D,32
)
1350  Y=Y+D
1360  CALL  SOUND(100,-8,5)
1370  CALL  HCHAR(X,Y,145-D)
1380  CALL  HCHAR(X,Y-D,147)
1390  CALL  HCHAR(X-1,Y-7*D,32
)
1400  GOTO  780
1410  IF  W=32  THEN  850
1420  CALL  HCHAR(24,W,32)
1430  W=W+1
1440  CALL  HCHAR(24,W,128)
1450  GOTO  850
1460  IF  W=2  THEN  850
1470  CALL  HCHAR(24,W,32)
1480  W=W-1
1490  CALL  HCHAR(24,W,128)
1500  GOTO  850
1510  CALL  SOUND(300,-7,0)
1520  CALL  GCHAR(V-1,W,U)
1530  CALL  HCHAR(V-1,W,152)
1540  CALL  HCHAR(V-1,W,32)
1550  IF  U=144  THEN  1630
1560  IF  U=145  THEN  1630
1570  IF  U=146  THEN  1630
1580  IF  U=147  THEN  1700
1590  IF  U=136  THEN  1760
1600  V=V-1
1610  IF  V=1  THEN  850
1620  GOTO  1520
1630  P=P+500
1640  CALL  SOUND(200,440,10)
1650  CALL  SOUND(200,550,10)
1660  CALL  SOUND(200,660,10)
1670  CALL  SOUND(400,880,10)
1680  V=24
1690  GOTO  850
1700  P=P+200
1710  CALL  SOUND(200,440,10)
1720  CALL  SOUND(200,660,10)
1730  CALL  SOUND(200,550,10)
1740  V=24
1750  GOTO  850
1760  P=P+100
1770  CALL  SOUND(200,660,10)
1780  CALL  SOUND(200,770,10)
1790  V=24
1800  GOTO  850
1810  CALL  SCREEN(15)
1820  IF  H<P  THEN  1830  ELSE  1
840
1830  H=P
1840  PRINT  :::"ENDE DES SPIE
LES"::"SIE HABEN"::P;"PUNKTE
 ERREICHT!":::"REKORD BISHER
:"::H;"PUNKTE":::
1850  PRINT  "EIN NEUES SPIEL?
 >>>  1"
1860  CALL  KEY(0,KEY,STATUS)
1870  IF  STATUS=0  THEN  1860
1880  IF  KEY=49  THEN  150  ELSE
 1860
1890
```

Frogpath

Dieses Spiel ist dem bekannten Spiel Frogger nachgebildet. Das Modul Extended Basic, sowie die Joysticks werden benötigt.

Sie haben 10 Frösche zur Verfügung. Ihre Aufgabe ist es, von diesen Fröschen mindestens 5 zu retten. Gesteuert wird mit Joystick Nr. 2. Zunächst überquert man eine vierspurige Straße - hier darf man nicht überfahren werden. Am Flußufer angekommen, muß man auf das Floß und die Tonnen springen, um den Fluß zu überqueren.
Schließlich gilt es noch, eine Landebahn für Flugzeuge zu überqueren, bis man den Frosch gerettet hat.
Dieses Spiel ist eines der schnellsten, die in TI Extended Basic möglich sind, da viele Aufgaben an Sprites übertragen wurden.
Es stehen insgesamt 40 Schwierigkeitsgrade zur Verfügung. Allerdings dürfte es hier niemand schaffen, auch nur einen Frosch zu retten.
Ab Runde 5 erscheint am Ufer eine Schlange, der man am Besten aus dem Weg geht. Ab runde 7 patrouillert auch am anderen Ufer eine Schlange.




```
1Ø REM ********************
2Ø REM PROGRAMMIERT VON
3Ø REM LUTZ STRADMANN
4Ø REM 3.5.1983
5Ø REM ALLE RECHTE VORBEHAL-
       TEN
6Ø REM ********************
7Ø CALL CLEAR::DISPLAY AT(12
,1Ø):"FROGPATH":: DISPLAY AT
(24,1):"SCHWIERIGKEITSGRAD?"
8Ø ACCEPT AT(24,2Ø)BEEP VALI
DATE(DIGIT) SIZE(2):V
9Ø DIM C(14)::CALL CLEAR::RA
NDOMIZE
1ØØ CALL DELSPRITE(ALL)
11Ø V=V-1::R=Ø::FROE=1Ø::RET
T=Ø::PUNKTE=Ø
12Ø CALL SCREEN(12)::CALL CO
LOR(9,14,14,1Ø,6,6,11,8,8,14
,16,1)
13Ø DISPLAY AT(1,1):"PUNKTE:
"
14Ø CALL MAGNIFY(3)
15Ø CALL HCHAR(2,1,97,32)::C
ALL HCHAR(22,1,97,32)!WAAGER
ECHTE UMRANDUNG
16Ø CALL HCHAR(21,1,1Ø4,32):
:CALL HCHAR(3,1,1Ø4,32)::CAL
L HCHAR(4,1,1Ø4,3)::CALL HCH
AR(4,5,1Ø5,5)::CALL HCHAR(4,
11,1Ø4,5)
```

```
17Ø CALL HCHAR(4,17,1Ø4,5)::
CALL HCHAR(4,23,1Ø4,5)::CALL
 HCHAR(4,29,1Ø4,4)
18Ø CALL HCHAR(7,1,1Ø4,32)::
 CALL HCHAR(12,1,1Ø4,32)::CA
LL HCHAR(8,1,112,128)!FLUSS
19Ø CALL CHAR(36,"ØØØØØØØ1Ø2
62E318ØØØØØØØØØØØØØØØØØØØ18
94D573Ø1ØØØØØØØØØØØØØØØØØ")
!SCHLANGE
2ØØ CALL CHAR(96,"ØØØØØØØ1Ø3
Ø6FFFFFFFF243C243C18ØØØØ7EFE
F6F6F6F6F6FEFE243C243C18ØØ")
!PKW
21Ø CALL CHAR(1Ø4,"ØØFFØØFFØ
ØFFØØFFØØFFØØFFØØFFØØFFØØFFØ
ØFFØØFFØØFFØØFFØØFFØØFFØØØØ"
)!FLOSS
22Ø CALL CHAR(112,"ØØ3F7FFFF
F7F3FØFØF3F7FFFFF7F3FØØØØFCF
EFFFFFEFCFØFØFCFEFFFFFEFØØØ"
)!TONNEN AUF DEM FLUSS
23Ø CALL CHAR(12Ø,"ØØØC1F372
727E7FFFFFF3F3FØFØ6ØØØØ1818F
CFEFEFFFFFFFFFFFFF1EØCØØØØØ
")!RETTUNGSWAGEN
24Ø CALL CHAR(124,"ØØØØØ3Ø7Ø
C183Ø7FFFFFFFFF4444EEEEØØØØC
ØEØ3Ø18ØCFEFFFFFFFF4444EEEE"
)!UFO
25Ø CALL CHAR(128,"ØØ7FØ17F7
D4D4D7D7F183C3C18ØØØØØFF839
3BB93939383FFØC1E1EØCØØØØ")!
LEICHENWAGEN
26Ø CALL CHAR(132,"ØØØØØØØØØØ
1Ø33FFFFF3FØ3Ø1ØØØØØØØØØØ3Ø7
ØFØF1F3FFFFFFFFF3F1FØ7Ø3ØØØ"
)!FLUGZEUG
27Ø CALL CHAR(136,"4224183C3
C3C4218ØØØØØØØØØØØØØØØØØØØØØ
ØØØØØØØØØØØØØØØ7ØØØØØØØØØØØØØ"
)!FROSCH
28Ø CALL SPRITE(#1,96,5,145,
256,#2,96,5,145,17Ø,#3,96,5,
145,85)!STRASSE 1.REIHE
29Ø CALL SPRITE(#4,124,7,129
,22Ø,#5,124,7,129,135,#6,124
,7,129,4Ø)!STRASSE 2.REIHE
3ØØ CALL SPRITE(#24,12Ø,14,1
13,175,#25,12Ø,14,113,1ØØ,#2
6,12Ø,14,113,1Ø)!STRASSE 3.R
EIHE
31Ø CALL SPRITE(#1Ø,128,9,98
,16Ø,#11,128,9,98,70,#12,128
,9,98,25Ø)!STRASSE 4.REIHE
32Ø CALL SPRITE(#2Ø,112,2,73
,25Ø,#21,112,2,73,1ØØ)!FLUSS
 1.REIHE
33Ø CALL SPRITE(#19,1Ø4,13,5
7,2ØØ,#18,1Ø4,13,57,12Ø,#17,
1Ø4,13,57,5Ø)!FLUSS 2.REIHE
34Ø CALL SPRITE(#15,132,8,33
,25Ø,#22,132,8,33,125)!LANDE
BAHN
35Ø GOTO 87Ø
36Ø R=Ø::CALL SOUND(1ØØ,789,
Ø)::CALL HCHAR(24,27,136)::I
F FROE=Ø THEN 1Ø1Ø ELSE DISP
LAY AT(24,26):FROE-1!NOCH VO
RHANDENE FROESCHE
37Ø ON ERROR 75Ø
38Ø X1=161::Y1=INT(RND*16Ø)+
4Ø!ANFANGSPOSITION FROSCH
39Ø CALL SPRITE(#16,136,16,X
1,Y1)
4ØØ CALL JOYST(2,Y,X)
41Ø IF X=4 THEN X1=X1-8 ELSE
 IF Y=4 THEN Y1=Y1+8 ELSE IF
 Y=-4 THEN Y1=Y1-8 ELSE IF X
=-4 AND X1> 16Ø THEN 4ØØ ELSE
 43Ø
42Ø GOTO 47Ø
43Ø IF X=-4 THEN X1=X1+8::GO
TO 47Ø
44Ø GOTO 45Ø
45Ø CALL COINC(ALL,B)::FC=FC
+1::IF B=-1 THEN 74Ø ELSE IF
 FC=16 THEN 46Ø ELSE 4ØØ
46Ø FC=Ø::H=-1*H::GF=-H::CAL
L MOTION(#28,Ø,H*(INT(RND*15
)+5))::CALL SOUND(-1,-5,Ø)::
CALL MOTION(#27,Ø,GF*(INT(RN
D*27)+6))::GOTO 4ØØ
47Ø IF X1<89 THEN 54Ø ELSE C
ALL LOCATE(#16,X1,Y1)::CALL
SOUND(-5,1ØØØ,5)::R=R+SGN(X)
::CALL COINC(ALL,B)::IF B=-1
 THEN 74Ø ELSE 4ØØ
48Ø IF X1=89 THEN CALL MOTIO
N(#16,Ø,Ø)::GOTO 47Ø!STEUERU
NG FUER FLUSS
49Ø CALL JOYST(2,X,Y)
5ØØ IF Y=4 THEN 54Ø ELSE IF
Y=-4 THEN 51Ø ELSE IF X=-4 T
HEN 57Ø ELSE IF X=4 THEN 59Ø
 ELSE 48Ø
51Ø CALL POSITION(#16,X1,Y1)
::R=R-1
52Ø X1=X1+8::CALL LOCATE(#16
,X1,Y1)::CALL MOTION(#16,Ø,C
(R))::CALL SOUND(-5,98Ø,5,99
5,6)::CALL COINC(ALL,B)::IF
B=-1 THEN 48Ø ELSE 53Ø
53Ø IF X1=89 THEN 48Ø ELSE 7
4Ø
54Ø CALL POSITION(#16,X1,Y1)
::R=R+1
55Ø X1=X1-8::CALL LOCATE(#16
```

```
,X1,Y1)::CALL MOTION(#16,Ø,C
(R))::CALL SOUND(-5,98Ø,5,99
5,6)::CALL COINC(ALL,B)::IF
B=-1 THEN 48Ø ELSE 56Ø
56Ø IF X1=49 THEN 6ØØ ELSE 7
4Ø
57Ø CALL POSITION(#16,X1,Y1)
::Y1=Y1-8
58Ø CALL LOCATE(#16,X1,Y1)::
CALL SOUND(-5,98Ø,5,995,6)::
CALL COINC(ALL,B)::IF B=Ø TH
EN 74Ø ELSE 48Ø
59Ø CALL POSITION(#16,X1,Y1)
::Y1=Y1+8::CALL LOCATE(#16,X
1,Y1)::CALL SOUND(-5,98Ø,5,9
95,6)::CALL COINC(ALL,B)::IF
B=Ø THEN 74Ø ELSE 48Ø
6ØØ CALL MOTION(#16,Ø,Ø)::CA
LL LOCATE(#16,INT(X1/8)*8+1,
INT(Y1+8)*8+8)::X1=INT(X1/8)
*8+1::Y1=INT(Y1/8)*8+8
61Ø !
62Ø CALL JOYST(2,X,Y)
63Ø IF X1=33 AND Y=4 THEN GO
SUB 68Ø ELSE IF Y=-4 AND X1=
49 THEN 49Ø
64Ø IF Y=4 THEN X1=X1-8 ELSE
 IF X=4 THEN Y1=Y1+8 ELSE IF
 X=-4 THEN Y1=Y1-8 ELSE IF Y
=-4 THEN X1=X1+8 ELSE 66Ø
65Ø GOTO 72Ø
66Ø CALL COINC(ALL,B)::IF B=
-1 THEN 74Ø
67Ø GOTO 62Ø
68Ø CALL GCHAR(4,INT(Y1/8)+1
,D)::IF D<>32 THEN 74Ø!SPRUN
G IN DEN SUMPF
69Ø PUNKTE=PUNKTE+2ØØ::DISPL
AY AT(1,9)SIZE(12):PUNKTE
7ØØ CALL POSITION(#16,X1,Y1)
:: CALL DELSPRITE(#16)::CALL
 HCHAR(X1/8,Y1/8,136)::GOTO
78Ø
71Ø RETURN
72Ø CALL LOCATE(#16,X1,Y1)::
CALL SOUND(-5,98Ø,5,995,6)::
CALL COINC(ALL,B)::IF B=-1 T
HEN 74Ø ELSE 61Ø
73Ø REM FROSCH WIRD UEBERFAH
REN ODER FAELLT IN FLUSS
74Ø CALL DELSPRITE(#16)::FOR
 X=1 TO 5::CALL SOUND(12Ø,18
Ø,3,19Ø,4)::CALL SOUND(24Ø,2
7Ø,3,28Ø,4)::NEXT X::FROE=FR
OE-1
75Ø IF FROE=Ø THEN 99Ø
76Ø FOR X=1 TO 35Ø::NEXT X::
GOTO 36Ø
77Ø REM FROSCH GERETTET
78Ø FOR Z=1 TO 1Ø::CALL SOUN
D(6Ø,56Ø,6,78Ø,7)::CALL SOUN
D(5Ø,98Ø,4,99Ø,6)::NEXT Z::R
ETT=RETT+1::IF RETT=5 THEN 8
1Ø
79Ø GOTO 36Ø
8ØØ REM 5 FROESCHE GERETTET
81Ø FOR X=1 TO 4::CALL SOUND
(12Ø,45Ø,2,46Ø,3)::CALL SOUN
D(12Ø,78Ø,3,79Ø,3)::CALL SOU
ND(12Ø,67Ø,3,68Ø,2)::NEXT X
82Ø DISPLAY BEEP AT(23,1):"G
RATULIERE,SIE HABEN 5 FROE-S
CHE GERETTET...VIEL GLUECK"
83Ø FROE=1Ø::RETT=Ø
84Ø FOR VERZ=1 TO 1ØØØ::NEXT
 VERZ::DISPLAY AT(23,1):"

"
85Ø V=V+1::IF V>4Ø THEN V=4Ø
 !BESCHLEUNIGUNG DER FAHRZEU
GE
86Ø DISPLAY AT(1,26):V::CALL
 SOUND(39Ø,78Ø,5,79Ø,6,77Ø,7
,-1,Ø)
87Ø CALL HCHAR(4,4,32)::CALL
 HCHAR(4,1Ø,32)::CALL HCHAR(
4,16,32)::CALL HCHAR(4,22,32
)::CALL HCHAR(4,28,32)
88Ø FOR X=1 TO 3::CALL MOTIO
N(#X,Ø,-(V+7))::NEXT X::FOR
X=4 TO 8::CALL MOTION(#X,Ø,V
+5)::NEXT X
89Ø FOR X=24 TO 26::CALL MOT
ION(#X,Ø,-(V+8))::NEXT X
9ØØ FOR X=1Ø TO 12::CALL MOT
ION(#X,Ø,V+9)::NEXT X
91Ø CALL MOTION(#15,Ø,-(V+18
),#22,Ø,-(V+18))
92Ø C(1Ø)=-(V+5)::C(11)=-(V+
5)::C(12)=V+7::C(13)=V+7
93Ø FOR X=2Ø TO 21::CALL MOT
ION(#X,Ø,C(10))::NEXT X::FOR
 X=17 TO 19::CALL MOTION(#X,
Ø,C(12))::NEXT X
94Ø IF V<5 THEN 97Ø ELSE H=I
NT(RND*2)::IF H=Ø THEN H=-1
95Ø CALL SPRITE(#28,36,15,89
,2ØØ,Ø,H*(INT(RND*22)+5))
96Ø IF V<7 THEN 97Ø ELSE GF=
-H::CALL SPRITE(#27,36,15,49
,5Ø,Ø,GF*(INT(RND*28)+5))
97Ø GOTO 36Ø
98Ø REM KEINE 5 FROESCHE GER
ETTET
99Ø DISPLAY AT(23,1):"NOCH E
IN SPIEL?(J/N)"::ACCEPT BEE
P AT(23,22):A$::IF A$="J" OR
 A$="j" THEN 9Ø ELSE STOP
```

Flugabwehrgeschütz

Das Spiel ist für den TI-99/4A gedacht und kann in der Grundversion (also ohne Extended Basic und Joysticks) benutzt werden. Der Sinn des Spieles liegt darin, auftauchende Flugzeuge mit einem Flugabwehr-Geschütz abzuschießen. Dieses befindet sich am unteren Bildschirmrand. Die Höhe und die Geschwindigkeit, in der Flugzeuge auftauchen kann man vor jedem Spielbeginn wählen (3 verschiedene Höhen, 4 verschiedene Geschwindigkeiten; also 12 Schwierigkeitsgrade).

Es tauchen immer 1 - 4 Flugzeuge zufällig in der gewählten Höhe auf. Allerdings ändert sich mit jedem Auftauchen der Flugzeuge auch die Position des Geschützes.

Geschossen wird mit der Leertaste. Nach 10 Fehlschüssen ist ein Spiel zu Ende.

```
LIST

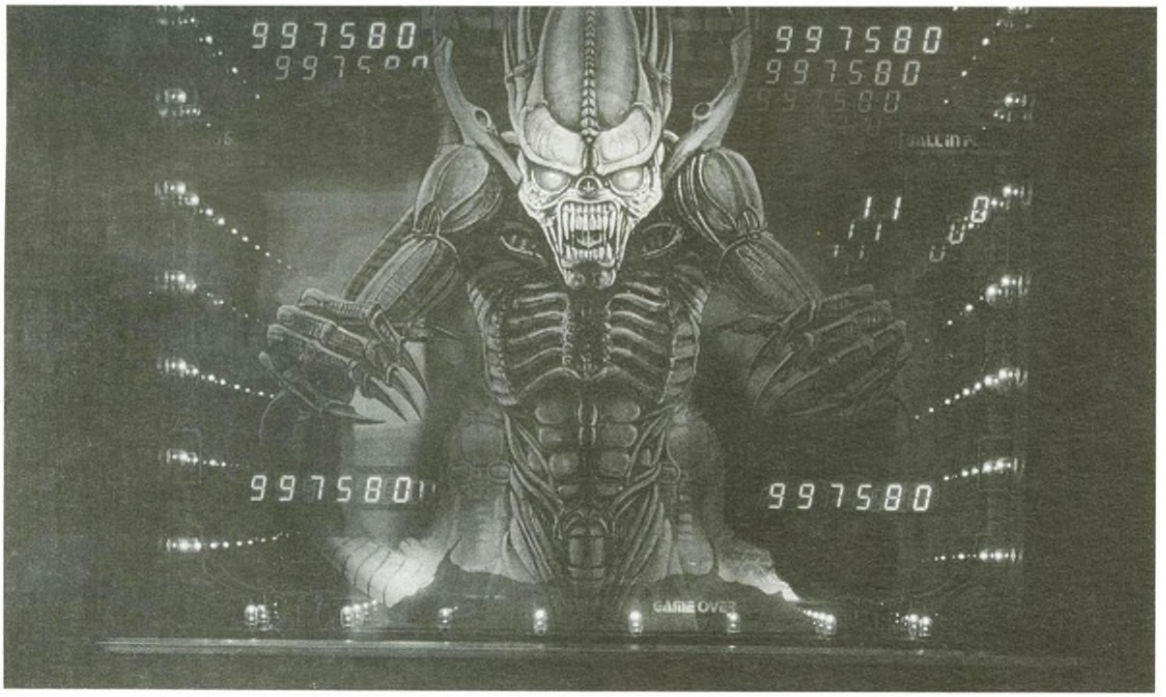
1ØØ HI=Ø
11Ø CALL CHAR(128,"FFFFFFFFF
FFFFFFF")
12Ø CALL CHAR(136,"1Ø1Ø98FFF
F981Ø1Ø")
13Ø CALL CHAR(144,"1818187E3
C3C3CFF")
14Ø CALL CHAR(152,"ØØ1818181
81818")
15Ø CALL CLEAR
16Ø GOSUB 148Ø
17Ø PRINT TAB(2);"DAS SPIEL
HAT DREI"::TAB(2);"SCHWIERIG
KEITSGRADE.":::::TAB(9);"1-L
EICHT"::TAB(9);"2-MITTEL"::
18Ø PRINT TAB(9);"3-SCHWER":
::::TAB(2);"WAEHLEN SIE NUN!
!!!":::::
19Ø PRINT TAB(2Ø);"V.LA.83"
2ØØ CALL SOUND(5ØØ,71Ø,Ø)
21Ø CALL KEY(3,KEY,STATUS)
22Ø IF STATUS=Ø THEN 21Ø
23Ø IF KEY=49 THEN 26Ø
24Ø IF KEY=5Ø THEN 3ØØ
25Ø IF KEY=51 THEN 33Ø ELSE
21Ø
26Ø CALL VCHAR(9,2Ø,42)
27Ø F=16
28Ø GOSUB 168Ø
29Ø GOTO 41Ø
3ØØ F=7
31Ø GOSUB 168Ø
32Ø GOTO 41Ø
33Ø F=3
34Ø GOSUB 168Ø
35Ø CALL CLEAR
36Ø H=1
37Ø I=1
38Ø ABCØ$="   TREFFER: Ø    F
EHLSCHUESSE: Ø"
39Ø GOSUB 119Ø
4ØØ GOTO 5ØØ
41Ø CALL CLEAR
42Ø H=1
43Ø I=1
44Ø ABCØ$="   TREFFER: Ø    F
EHLSCHUESSE: Ø"
45Ø GOSUB 119Ø
46Ø CALL HCHAR(2,2,128,31)
47Ø CALL HCHAR(24,2,128,31)
48Ø CALL VCHAR(2,32,128,23)
49Ø CALL VCHAR(2,2,128,23)
5ØØ RANDOMIZE
51Ø E=INT(26*RND)+5
52Ø CALL HCHAR(23,E,144)
53Ø RANDOMIZE
54Ø X=INT(26*RND)+5
55Ø RANDOMIZE
56Ø A=INT(26*RND)+5
57Ø RANDOMIZE
58Ø B=INT(26*RND)+5
59Ø RANDOMIZE
6ØØ C=INT(26*RND)+5
61Ø CALL HCHAR(F,X,136)
62Ø CALL HCHAR(F,A,136)
63Ø CALL HCHAR(F,B,136)
64Ø CALL HCHAR(F,C,136)
65Ø GOTO 72Ø
66Ø CALL HCHAR(F,X,32)
67Ø CALL HCHAR(F,A,32)
68Ø CALL HCHAR(F,B,32)
69Ø CALL HCHAR(F,C,32)
7ØØ CALL HCHAR(23,E,32)
71Ø GOTO 5ØØ
72Ø FOR Y=1 TO CA
73Ø IF CB=1 THEN 79Ø
74Ø CALL KEY(3,KEY,STATUS)
75Ø IF STATUS=Ø THEN 79Ø
76Ø IF KEY=32 THEN 77Ø ELSE
79Ø
77Ø GOSUB 82Ø
78Ø CB=CB+1
79Ø NEXT Y
8ØØ CB=Ø
81Ø GOTO 66Ø
```

```
82Ø FOR G=22 TO F STEP -1
83Ø CALL SOUND(1ØØ,-3,Ø)
84Ø CALL HCHAR(G,E,152)
85Ø NEXT G
86Ø IF X=E THEN 1ØØØ
87Ø IF A=E THEN 1ØØØ
88Ø IF B=E THEN 1ØØØ
89Ø IF C=E THEN 1ØØØ
900 R=R+1
91Ø S=R+48
92Ø CALL HCHAR(1,32,S)
93Ø IF Q>HI THEN 94Ø ELSE 95
Ø
94Ø HI=Q
95Ø IF R=1Ø THEN 125Ø
96Ø FOR G=F TO 22
97Ø CALL HCHAR(G,E,32)
98Ø NEXT G
99Ø RETURN
1ØØØ Q=Q+1
1Ø1Ø IF Q>999 THEN 1Ø2Ø ELSE
 1Ø3Ø
1Ø2Ø Q=Ø
1Ø3Ø CALL SOUND(3ØØ,415,Ø)
1Ø4Ø CALL SOUND(3ØØ,622,Ø)
1Ø5Ø CALL SOUND(3ØØ,523,Ø)
1Ø6Ø AA=INT(Q/1ØØ)
1Ø7Ø AB=INT((Q-INT(Q/1ØØ)*1Ø
Ø)/1Ø)
1Ø8Ø AC=Q-AA*1ØØ-AB*1Ø
1Ø9Ø AA=AA+48
11ØØ AB=AB+48
111Ø AC=AC+48
112Ø CALL HCHAR(1,13,AA)
113Ø CALL HCHAR(1,14,AB)
114Ø CALL HCHAR(1,15,AC)
115Ø FOR G=F TO 22
116Ø CALL HCHAR(G,E,32)
117Ø NEXT G
118Ø RETURN
119Ø FOR J=1 TO LEN(ABC$)
12ØØ CO=ASC(SEG$(ABC$,J,1))
121Ø CALL HCHAR(H,I,CO)
122Ø I=I+1
123Ø NEXT J
124Ø RETURN
125Ø CALL CLEAR
126Ø IF HI=Q THEN 127Ø ELSE
131Ø
127Ø CALL SOUND(3ØØ,415,Ø)
128Ø CALL SOUND(3ØØ,311,Ø)
129Ø CALL SOUND(3ØØ,523,Ø)
13ØØ CALL SOUND(3ØØ,415,Ø)
131Ø PRINT TAB(8);"HOECHSTER
GEBNIS :";HI:::::::
132Ø PRINT TAB(9);"SPIELENDE
 !!":::::"  SIE HABEN";Q;"FLU
GZEUGE"::"  GETROFFEN"::::
133Ø PRINT "  WOLLEN SIE NOC
HMAL"::"  SPIELEN ? <J><N>"
134Ø IF Q=1 THEN 135Ø ELSE 1
36Ø
135Ø CALL HCHAR(15,25,32)
136Ø R=Ø
137Ø Q=Ø
138Ø CALL SOUND(5ØØ,71Ø,Ø)
139Ø CALL KEY(3,KEY,STATUS)
14ØØ IF STATUS=Ø THEN 139Ø
141Ø IF KEY=74 THEN 11Ø
142Ø IF KEY=78 THEN 143Ø ELS
E 139Ø
143Ø CALL CLEAR
144Ø PRINT TAB(7);"COPYRIGHT
"::TAB(7);"BY VOLKER LAMEK":
:TAB(7);"1983"::::::::
145Ø FOR X=1 TO 6ØØØ
146Ø NEXT X
147Ø END
148Ø CALL SCREEN(5)
149Ø CALL COLOR(1,4,5)
15ØØ CALL COLOR(2,4,5)
151Ø CALL COLOR(3,4,5)
152Ø CALL COLOR(4,4,5)
153Ø CALL COLOR(5,4,5)
154Ø CALL COLOR(6,4,5)
155Ø CALL COLOR(7,4,5)
156Ø CALL COLOR(8,4,5)
157Ø CALL COLOR(9,4,5)
158Ø CALL COLOR(13,1Ø,5)
159Ø CALL COLOR(14,12,5)
16ØØ CALL COLOR(15,2,5)
161Ø CALL COLOR(16,16,5)
162Ø RETURN
168Ø CALL CLEAR
169Ø PRINT "  WAEHLEN SIE NU
N DIE"::"  GESCHWINDIGKEIT,I
N DER "::"  DIE FLUGZEUGE AU
FTAUCHEN"::"  SOLLEN !"::::
17ØØ PRINT TAB(9);"1) LANGSA
M"::TAB(9);"2) SCHNELL"::TAB
(9);"3) SCHNELLER"::TAB(9);"
4) AM SCHNELLSTEN"::::::
171Ø CALL SOUND(5ØØ,7ØØ,Ø)
172Ø CALL KEY(3,KEY,STATUS)
173Ø IF STATUS=Ø THEN 172Ø
174Ø IF KEY=49 THEN 184Ø
175Ø IF KEY=5Ø THEN 182Ø
176Ø IF KEY=51 THEN 18ØØ
177Ø IF KEY=52 THEN 178Ø ELS
E 172Ø
178Ø CA=4
179Ø RETURN
18ØØ CA=6
181Ø RETURN
182Ø CA=8
183Ø RETURN
184Ø CA=15
185Ø RETURN
```

Monster Hunt

Monster Hunt hat eine gewisse Ähnlichkeit mit Pacman, die jedoch wirklich nur sehr entfernt ist.

Das Programm läuft auf der Grundversion des TI-99 ohne Joystick. Es kann dabei in vier Richtungen gesteuert werden. Dies geschieht mit den Tasten 1 bis 4 (1=links, 2=rechts, 3=hoch, 4=runter. Steuert man 1 und 2 mit zwei Fingern der linken Hand und 3 und 4 mit zwei Fingern der rechten Hand, so steuert man nach kurzer Gewöhnungszeit die Zielmarke, die in diesem Spiel die Form einer Raute hat, fast ebenso schnell, wie mit einer Fernbedienung.
Nun zum Spiel und seinen Regeln: Zu Anfang hat der Spieler 500 Energieeinheiten. Wird die Zielmarke bewegt, so benötigt man zwei Einheiten für jeden Schritt; steht sie, so zählt der Computer kontinuierlich um 1 rückwärts. Man kann zu Anfang durch die Gänge marschieren und die blauen Punkte löschen. Jede Löschung bringt vier Punkte ein. Von Zeit zu Zeit (vom Zufallszahlen-Generator bestimmt, ertönt ein tiefer Ton und die Fortbewegung wird gestoppt, damit der Computer 6 neue Monster plazieren kann.
Nun heißt es schnell sein, denn wenn man zu lange wartet, verschwinden die Monster wieder. Man muß also die gelben, kleineren Monster schnell erreichen und vom Feld löschen, was jeweils 100 Punkte einbringt, den roten jedoch nicht zu nahe kommen: weil sie ungenießbar sind und die Marke des Spielers verschlingen. Sollten die Monster einmal zuviel werden, so kann man auch auf einer Stelle kurz stehenbleiben und warten. damit die zuletzt aufgetauchten wieder verschwinden. Ideal ist es natürlich, so schnell wie möglich alle gelben zu löschen und die roten durch kurzes Warten wieder verschwinden lassen. Wenn einmal rote Monster alle Ausgänge versperren sollten, so kann man auch ein Mauerstück einreißen, was 200 Energie-Einheiten kostet. Überhaupt sollte man bei seiner Wanderung durch die verschiedenen Gänge aufpassen, daß die Energie nicht zu wenig wird. Nur wenn man den Energiespeicher in der Mitte schnell genug erreicht, kann man wieder suf 500 Energieeinheiten auftanken,wobei jedesmal ein Stück des Spielers verlorengeht. Wenn man allerdings mit seiner Energie sparsam umgeht und den roten Monstern ausweicht, spielt man an einer Runde u.U. 15 bis 20 Minuten. Übrigens: Punktzahl, Hihscore und Energievorrat werden ständig auf dem Bildschirm angezeigt; und das Ganze ist natürlich mit Soundeffekten untermalt.




```
1Ø REM  **************
2Ø REM  *MONSTER-HUNT*
3Ø REM  *            *
4Ø REM  * DEVELOPED  *
5Ø REM  * BY A.ERTL  *
6Ø REM  **************
1ØØ RANDOMIZE
11Ø CALL CLEAR
12Ø CALL SCREEN(16)
13Ø FOR ZS=1 TO 4
14Ø CALL COLOR(ZS,5,1)
15Ø NEXT ZS
16Ø FOR ZS=5 TO 8
17Ø CALL COLOR(ZS,9,1)
18Ø NEXT ZS
19Ø CALL CHAR(115,"182442818
    1422418")
2ØØ CALL CHAR(12Ø,"ØØ183C7E5
```

```
     A7E7E7E")
210 CALL CHAR(130,"183C7EDBF
     FC3DBFF")
220 CALL CHAR(140,"0000183C3
     C18")
230 CALL CHAR(150,"FFC381818
     181C3FF")
240 CALL CHAR(152,"000000FFF
     F")
250 CALL CHAR(153,"181818181
     8181818")
260 CALL CHAR(154,"000000F8F
     8181818")
270 CALL CHAR(155,"000000 1F1
     F181818")
280 CALL CHAR(156,"1818181F1
     F")
290 CALL CHAR(157,"181818F8F
     8")
300 CALL COLOR(11,5,1)
310 CALL COLOR(12,12,1)
320 CALL COLOR(13,10,1)
330 CALL COLOR(14,8,1)
340 CALL COLOR(15,3,1)
350 CALL COLOR(16,13,1)
360 CALL CLEAR
370 VT=1
380 HT=3
390 PRI$="PKT"
400 GOSUB 1970
410 HT=13
420 PRI$="HISC"
430 GOSUB 1970
440 HT=24
450 PRI$="EN"
460 GOSUB 1970
470 RESTORE
480 FOR I=1 TO 22
490 READ A,B,C
500 CALL HCHAR(A,B,152,C)
510 NEXT I
520 DATA 3,4,26,23,4,26,5,6,
     3,5,11,5,5,18,5,5,25,3
530 DATA 21,6,3,21,11,5,21,1
     8,5,21,25,3,7,8,18,19,8,
     18
540 DATA 9,10,2,9,13,6,9,22,
     2,17,10,2,17,13,6,17,22,
     2,11,12,4,11,18,4,15,18,
     4,15,12,4
550 FOR I=1 TO 14
560 READ A,B,C
570 CALL VCHAR(A,B,153,C)
580 NEXT I
590 DATA 4,3,19,4,30,19,6,5,
     15,6,28,15,8,7,2,12,7,3,
     17,7,2,8,26,2,12,26,2,17
     ,26,2
600 DATA 12,11,3,12,22,3,10,
     9,7,10,24,7
610 FOR I=1 TO 20
620 READ A,B,C
630 CALL HCHAR(A,B,C)
640 NEXT I
650 DATA 3,3,155,3,30,154,23
     ,3,156,23,30,157,5,5,155
     ,5,28,154,21,5,156,21,28
     ,157
660 DATA 7,7,155,7,26,154,19
     ,7,156,19,26,157,9,9,155
     ,9,24,154,17,9,156,17,24
     ,157
670 DATA 11,11,155,11,22,154
     ,15,11,156,15,22,157
680 FOR I=1 TO 4
690 READ A,B,C,D,E,F
700 CALL HCHAR(A,B,140,C)
710 CALL VCHAR(D,E,140,F)
720 NEXT I
730 DATA 4,4,26,4,4,19,22,4,
     26,4,29,19,8,8,18,8,8,11
     ,18,8,18,8,25,11
740 FOR I=12 TO 14
750 CALL HCHAR(I,12,150,10)
760 NEXT I
770 P=0
780 EN=500
790 PA=10
800 PB=16
810 GOSUB 1820
820 GOSUB 1860
830 GOSUB 1890
840 DIM MA(6),MB(6)
850 CONT=0
860 ZAEHL=0
870 CALL HCHAR(PA,PB,115)
880 CALL SOUND(50,600,4)
890 CALL KEY(0,K,ST)
900 IF ST THEN 1120
910 EN=EN-1
920 IF EN<0 THEN 1640
930 GOSUB 1890
940 ZAEHL=ZAEHL+1
950 IF ZAEHL=5 THEN 2070
960 IF RND<.98 THEN 870
970 CALL SOUND(50,200,3)
980 FOR I=1 TO 6
990 MA(I)=INT(19*RND)+4
1000 MB(I)=INT(26*RND)+4
1010 CALL GCHAR(MA(I),MB(I),
     D)
1020 IF D=32 THEN 1040
1030 IF D>149 THEN 990
1040 NEXT I
1050 CALL SOUND(50,200,3)
1060 FOR I=1 TO 6
1070 CALL HCHAR(MA(I),MB(I),
     (INT(2*RND)+12)*10)
1080 CALL SOUND(50,300,3)
1090 NEXT I
```

```
1100 CONT=1
1110 GOTO 870
1120 CALL HCHAR(PA,PB,32)
1130 ON K-48 GOTO 1140,1180,
     1220,1260
1140 PB=PB-1
1150 IF PB>1 THEN 1290
1160 PB=32
1170 GOTO 1290
1180 PB=PB+1
1190 IF PB<32 THEN 1290
1200 PB=1
1210 GOTO 1290
1220 PA=PA-1
1230 IF PA>1 THEN 1290
1240 PA=24
1250 GOTO 1290
1260 PA=PA+1
1270 IF PA<24 THEN 1290
1280 PA=1
1290 EN=EN-2
1300 IF EN 0 THEN 1640
1310 GOSUB 1890
1320 CALL GCHAR(PA,PB,E)
1330 CALL HCHAR(PA,PB,115)
1340 IF E=32 THEN 1560
1350 IF E=150 THEN 1520
1360 IF E=130 THEN 1570
1370 IF E=120 THEN 1490
1380 IF E<>140 THEN 1430
1390 CALL SOUND(50,1000,4)
1400 P=P+4
1410 GOSUB 1820
1420 GOTO 870
1430 CALL SOUND(200,110,2)
1440 WAND=1
1450 EN=EN-200
1460 IF EN<0 THEN 1640
1470 GOSUB 1890
1480 GOTO 870
1490 P=P+100
1500 GOSUB 1820
1510 GOTO 870
1520 CALL SOUND(1000,2000,2)
1530 EN=500
1540 GOSUB 1890
1550 GOTO 870
1560 IF RND>.95 THEN 970 ELS
     E 870
1570 CALL CLEAR
1580 FOR ZS=1 TO 8
1590 CALL COLOR(ZS,INT(13*RN
     D+2,1)
1600 NEXT ZS
1610 PRINT TAB(10);"GAME OVE
     R"::::
1620 PRINT "DU WURDEST VERSC
     HLUNGEN.":::::
1630 GOTO 1660
1640 CALL CLEAR
1650 PRINT "DEINE ENERGIE IS
     T ZU ENDE.":::::::
1660 PRINT "PUNKTE:";P
1670 IF P<H THEN 1700
1680 GOSUB 2010
1690 H=P
1700 PRINT "HI-SCORE:";H:::
1710 PRINT "NOCHMAL?  /J/ /N
     /"
1720 CALL KEY(0,K,ST)
1730 IF ST=0 THEN 1720
1740 IF K=74 THEN 100
1750 IF K=78 THEN 1780
1760 CALL SOUND(300,110,2)
1770 GOTO 1720
1780 CALL CLEAR
1790 PRINT "NEUSTART MIT ""R
     UN"".":::::::
1800 PRINT "****************
     ************":::
1810 END
1820 FOR UI=1 TO LEN(STR$(P)
     )
1830 CALL HCHAR(1,6+UI,ASC(S
     EG$(STR$(P),UI,1)))
1840 NEXT UI
1850 RETURN
1860 FOR UI=1 TO LEN(STR$(H)
     )
1870 CALL HCHAR(1,17+UI,ASC(
     SEG$(STR$(H),UI,1)))
1880 NEXT UI
1890 EN$=STR$(EN)
1900 IF LEN(EN$)>2 THEN 1930
1910 EN$="0"&EN$
1920 GOTO 1900
1930 FOR UI=1 TO 3
1940 CALL HCHAR(1,26+UI,ASC(
     SEG$(EN$,UI,1)))
1950 NEXT UI
1960 RETURN
1970 FOR UI=1 TO LEN(PRI$)
1980 CALL HCHAR(VT,HT+UI-1,A
     SC(SEG$(PRI$,UI,1)))
1990 NEXT UI
2000 RETURN
2010 FOR UI=1 TO 6
2020 FOR UJ=-3 TO -1
2030 CALL SOUND(-100,UJ,1)
2040 NEXT UJ
2050 NEXT UI
2060 RETURN
2070 IF CONT=0 THEN 850
2080 FOR UI=1 TO 6
2090 CALL HCHAR(MA(UI),MB(UI
     ),32)
2100 CALL SOUND(50,400,3)
2110 NEXT UI
2120 CONT=0
2130 GOTO 850
```

Spielautomat TI-99/4A

Die erste Scheibe kann in dem Zeitraum in dem die Schrift "Start" erscheint, durch Betätigung der Taste "S", einmal nachgestartet werden. Wird im Ablauf des Spieles ein Geldbetrag oder Sonderspiel gewonnen, so kann man diesen Gewinn, durch Betätigung der Taste "R" (doppelt oder nichts) riskieren. Beim Sonderspielstand 6 und 11 können, bei Kronensymbol im Mittelfenster, weitere Sonderspiele gewonnen werden. Der Münzspeicher zeigt auch negative Beträge an, so daß man zu jedem Zeitpunkt des Spieles erkennen kann, wieviel man insgesamt gewonnen oder verloren hat.

```
100 REM  SPIELAUTOMAT
110 REM  FARBEN
120 CALL COLOR(9,12,6)
130 REM  ZEICHENDEFINITION
140 REM  KRONE
150 CALL CHAR(96,"00100C0E07070303")
160 CALL CHAR(97,"0000187EFFFFFFFF")
170 CALL CHAR(98,"00107060E0E0C0C0")
180 CALL CHAR(99,"070703030303")
190 CALL CHAR(100,"FFFFFFFFFFFF")
200 CALL CHAR(101,"E0E0C0C0C0C0")
210 REM  ERSTELLUNG VON DATENFELDERN
         FUER WERTZUWEISUNG
220 DIM WER(12)
230 FOR Q=1 TO 12
240 READ WER(Q)
250 NEXT Q
260 DATA 77,8,4,24,6,12,6,8,4,30,6,16
270 DIM WERT(2,12)
280 FOR A=1 TO 2
290 FOR B=1 TO 12
300 READ WERT(A,B)
310 NEXT B
320 NEXT A
330 DATA 4,6,8,12,6,4,8,12,6,4,6,12,16,24,
    30,77,16,24,30,77,16,24,30,77
340 MUENZSPEICHER=0
350 REM  BILDAUFBAU
360 CALL CLEAR
370 CALL CHAR(120,"FFFFFFFFFFFFFFFF")
380 CALL COLOR(12,7,7)
390 CALL HCHAR(3,4,120,7)
400 CALL HCHAR(3,22,120,7)
410 CALL VCHAR(4,4,120,4)
420 CALL VCHAR(4,10,120,4)
430 CALL VCHAR(4,22,120,4)
440 CALL VCHAR(3,28,120,5)
450 CALL HCHAR(7,5,120,5)
460 CALL HCHAR(7,13,120,7)
470 CALL HCHAR(7,23,120,5)
480 CALL VCHAR(8,13,120,3)
490 CALL VCHAR(8,19,120,3)
500 CALL HCHAR(11,4,120,7)
510 CALL HCHAR(11,13,120,7)
520 CALL HCHAR(11,22,120,7)
530 CALL VCHAR(12,4,120,3)
540 CALL VCHAR(12,10,120,3)
550 CALL VCHAR(12,22,120,3)
560 CALL VCHAR(12,28,120,3)
570 CALL HCHAR(15,4,120,7)
580 CALL HCHAR(15,22,120,7)
590 CALL COLOR(11,4,4)
600 CALL CHAR(117,"FFFFFFFFFFFFFFFF"
610 CALL HCHAR(12,6,117,3)
620 CALL HCHAR(12,24,117,3)
630 CALL HCHAR(8,15,117,3)
640 VAR=0
650 RANDOMIZE
660 VAR=VAR+1
670 DREH1=INT(12*RND)+1
680 X1=WERT(1,DREH1)
690 X2=WERT(2,DREH1)
700 Z=5
710 S=6
720 IF X1=4 THEN 830
730 IF X1=6 THEN 850
740 IF X1=8 THEN 870
750 IF X1=12 THEN 890
760 Z=13
770 S=6
780 CALL HCHAR(Z-1,S,115,3)
790 IF X2=16 THEN 910
800 IF X2=24 THEN 930
810 IF X2=30 THEN 950
820 IF X2=77 THEN 970
830 GOSUB 1720
840 GOTO 760
850 GOSUB 1760
860 GOTO 760
870 GOSUB 1800
880 GOTO 760
890 GOSUB 1840
900 GOTO 760
910 GOSUB 1880
920 GOTO 980
930 GOSUB 1920
940 GOTO 980
950 GOSUB 1960
960 GOTO 980
970 GOSUB 2000
980 REM  NACHSTART VON SCHEIBE1
990 IF VAR>1 THEN 1100
1000 CALL SOUND(50,440,1)
1010 CALL HCHAR(18,5,83)
1020 CALL HCHAR(18,6,84)
1030 CALL HCHAR(18,7,65)
1040 CALL HCHAR(18,8,82)
1050 CALL HCHAR(18,9,84)
1060 FOR START=1 TO 50
1070 CALL KEY(0,KEY,STATUS)
1080 IF KEY=83 THEN 650
1090 NEXT START
1100 CALL HCHAR(18,5,115,5)
1110 RANDOMIZE
1120 DREH2=INT(12*RND)+1
1130 X3=WERT(1,DREH2)
1140 X4=WERT(2,DREH2)
1150 Z=5
1160 S=24
1170 IF X3=4 THEN 1270
1180 IF X3=6 THEN 1290
1190 IF X3=8 THEN 1310
```

```
1200 IF X3=12 THEN 1330
1210 Z=13
1220 S=24
1230 IF X4=16 THEN 1350
1240 IF X4=24 THEN 1370
1250 IF X4=30 THEN 1390
1260 IF X4=77 THEN 1410
1270 GOSUB 1720
1280 GOTO 1210
1290 GOSUB 1760
1300 GOTO 1210
1310 GOSUB 1800
1320 GOTO 1210
1330 GOSUB 1840
1340 GOTO 1210
1350 GOSUB 1880
1360 GOTO 1420
1370 GOSUB 1920
1380 GOTO 1420
1390 GOSUB 1960
1400 GOTO 1420
1410 GOSUB 2000
1420 REM  DREHUNG MITTE
1430 RANDOMIZE
1440 DREH3=INT(12*RND)+1
1450 X5=WER(DREH3)
1460 Z=9
1470 S=15
1480 IF X5=4 THEN 1560
1490 IF X5=6 THEN 1580
1500 IF X5=8 THEN 1600
1510 IF X5=12 THEN 1620
1520 IF X5=16 THEN 1640
1530 IF X5=24 THEN 1660
1540 IF X5=30 THEN 1680
1550 IF X5=77 THEN 1700
1560 GOSUB 1720
1570 GOTO 2070
1580 GOSUB 1760
1590 GOTO 2070
1600 GOSUB 1800
1610 GOTO 2070
1620 GOSUB 1840
1630 GOTO 2070
1640 GOSUB 1880
1650 GOTO 2070
1660 GOSUB 1920
1670 GOTO 2070
1680 GOSUB 1960
1690 GOTO 2070
1700 GOSUB 2000
1710 GOTO 2070
1720 CALL HCHAR(Z,S,48)
1730 CALL HCHAR(Z,S+1,52)
1740 CALL HCHAR(Z,S+2,48)
1750 RETURN
1760 CALL HCHAR(Z,S,48)
1770 CALL HCHAR(Z,S+1,54)
1780 CALL HCHAR(Z,S+2,48)
1790 RETURN
1800 CALL HCHAR(Z,S,48)
1810 CALL HCHAR(Z,S+1,56)
1820 CALL HCHAR(Z,S+2,48)
1830 RETURN
1840 CALL HCHAR(Z,S,49)
1850 CALL HCHAR(Z,S+1,50)
1860 CALL HCHAR(Z,S+2,48)
1870 RETURN
1880 CALL HCHAR(Z,S,49)
1890 CALL HCHAR(Z,S+1,54)
1900 CALL HCHAR(Z,S+2,48)
1910 RETURN
1920 CALL HCHAR(Z,S,50)
1930 CALL HCHAR(Z,S+1,52)
1940 CALL HCHAR(Z,S+2,48)
1950 RETURN
1960 CALL HCHAR(Z,S,51)
1970 CALL HCHAR(Z,S+1,48)
1980 CALL HCHAR(Z,S+2,48)
1990 RETURN
2000 CALL VCHAR(Z-1,S,96)
2010 CALL VCHAR(Z-1,S+1,97)
2020 CALL VCHAR(Z-1,S+2,98)
2030 CALL VCHAR(Z,S,99)
2040 CALL VCHAR(Z,S+1,100)
2050 CALL VCHAR(Z,S+2,101)
2060 RETURN
2070 FOR T=1 TO 300
2080 NEXT T
2090 REM  SCORING
2100 IF SON>0 THEN 3670
2110 IF X2=X4 THEN 2200
2120 IF X1=X3 THEN 2160
2130 IF X5=77 THEN 2270
2140 VALUE=0
2150 GOTO 2370
2160 IF X1=X5 THEN 2250
2170 IF X5=77 THEN 2250
2180 VALUE=0
2190 GOTO 2370
2200 IF X2=77 THEN 2310
2210 IF X2=X5 THEN 2290
2220 IF X5=77 THEN 2290
2230 VALUE=0
2240 GOTO 2120
2250 VALUE=X1/10
2260 GOTO 2430
2270 VALUE=0.30
2280 GOTO 2430
2290 VALUE=X2/10
2300 GOTO 2430
2310 IF X5=77 THEN 3150
2320 IF X5<>12 THEN 2350
2330 VALUE=1.2
2340 GOTO 2430
2350 VALUE=0
2360 CALL CLEAR
2370 MUENZSPEICER=MUENZSPEICER+VALUE-0.30
2380 PRINT "SONDERSPIELE=";SON
2390 PRINT "MUENZSPEICHER=";MUENZSPEICER
2400 FOR T=1 TO 500
2410 NEXT T
2420 GOTO 350
2430 REM  RISIKO
2440 IF ANZAHL>48 THEN 2510
2450 FOR U=1 TO 50
2460 CALL KEY(0,KEY,STATUS)
2470 CALL SOUND(100,660,1)
2480 IF KEY=82 THEN 2580
2490 NEXT U
2500 IF ANZAHL=0 THEN 2540
2510 PRINT "KEIN RISIKO","GEWINN="
     ;ANZAHL;"SONDERSPIELE"
2520 GOSUB 3290
2530 GOTO 2550
2540 PRINT "KEIN RISIKO","GEWINN=";VALUE
2550 FOR T=1 TO 500
2560 NEXT T
```

```
2570 GOTO 2360
2580 RANDOMIZE
2590 RISK=2*RND
2600 IF RISK<1 THEN 2680
2610 VALUE=0
2620 ANZAHL=0
2630 PRINT "VERLOREN"
2640 CALL SOUND(-500,440,1)
2650 CALL SOUND(-500,343,1)
2660 CALL SOUND(2000,220,1,330,1,532,1)
2670 GOTO 2360
2680 IF VALUE<>.30 THEN 2720
2690 VALUE=0.60
2700 GOSUB 3060
2710 GOTO 2430
2720 IF VALUE<>0.40 THEN 2760
2730 VALUE=0.80
2740 GOSUB 3060
2750 GOTO 2430
2760 IF VALUE<>0.60 THEN 2800
2770 VALUE=1.20
2780 GOSUB 3060
2790 GOTO 2430
2800 IF VALUE<>0.80 THEN 2840
2810 VALUE=1.60
2820 GOSUB 3060
2830 GOTO 2430
2840 IF VALUE<>1.20 THEN 2880
2850 VALUE=2.40
2860 GOSUB 3060
2870 GOTO 2430
2880 IF VALUE<>1.60 THEN 2920
2890 VALUE=3.00
2900 GOSUB 3060
2910 GOTO 2430
2920 IF VALUE<>2.40 THEN 2970
2930 ANZAHL=3
2940 VALUE=0
2950 GOSUB 3060
2960 GOTO 2430
2970 IF VALUE<>3 THEN 3020
2980 ANZAHL=5
2990 VALUE=0
3000 GOSUB 3060
3010 GOTO 2430
3020 ANZAHL=ANZAHL*2
3030 GOSUB 3060
3040 GOTO 2430
3050 REM  SOUND BEI GEWINN
3060 CALL SOUND(-1000,220,0)
3070 CALL SOUND(-1000,277,0)
3080 CALL SOUND(-1000,330,0)
3090 CALL SOUND(-740,440,0,659,0,1109,0)
3100 IF VALUE=0 THEN 3130
3110 PRINT "GEWINN=";VALUE
3120 GOTO 3140
3130 PRINT "GEWINN";ANZAHL;"SONDERSPIELE"
3140 RETURN
3150 REM  SONDERSPIELAUSLOSUNG
3160 VALUE=0
3170 RANDOMIZE
3180 CALL CLEAR
3190 FOR T=1 TO 25
3200 GOSUB 3410
3210 CALL SOUND(50,TONE,0)
3220 PRINT ANZAHL
3230 NEXT T
3240 CALL SOUND(300,-3,0)
3250 FOR W=1 TO 24
3260 PRINT ANZAHL
3270 NEXT W
3280 GOTO 3380
3290 A=110
3300 FOR ZAEHL=1 TO ANZAHL
3310 SON=SON+1
3320 PRINT "SONDERSPIELZAEHLER=";SON
3330 CALL SOUND(50,A,1)
3340 A=A+10
3350 NEXT ZAEHL
3360 ANZAHL=0
3370 RETURN
3380 IF ANZAHL<50 THEN 2430
3390 GOSUB 3290
3400 GOTO 360
3410 TONE=INT(110*(8*RND+1))
3420 ZUFALL=100*RND+1
3430 IF ZUFALL>97 THEN 3520
3440 IF ZUFALL>93 THEN 3540
3450 IF ZUFALL>87 THEN 3560
3460 IF ZUFALL>80 THEN 3580
3470 IF ZUFALL>70 THEN 3600
3480 IF ZUFALL>55 THEN 3620
3490 IF ZUFALL>35 THEN 3640
3500 ANZAHL=3
3510 RETURN
3520 ANZAHL=100
3530 RETURN
3540 ANZAHL=50
3550 RETURN
3560 ANZAHL=40
3570 RETURN
3580 ANZAHL=20
3590 RETURN
3600 ANZAHL=10
3610 RETURN
3620 ANZAHL=6
3630 RETURN
3640 ANZAHL=5
3650 RETURN
3660 REM  SONDERSPIELWERTUNG
3670 IF SON=6 THEN 3850
3680 IF SON=11 THEN 3850
3690 IF X5=4 THEN 3750
3700 IF X5=6 THEN 3750
3710 IF X5=77 THEN 3750
3720 VALUE=0
3730 SON=SON-1
3740 GOTO 2360
3750 VALUE=3.00
3760 CALL SOUND(-500,220,1)
3770 CALL SOUND(-500,277,1)
3780 CALL SOUND(-500,330,1)
3790 CALL SOUND(-740,440,0,659,0,1109,0)
3800 SON=SON-1
3810 IF SON=1 THEN 3830
3820 IF SON=0 THEN 3830 ELSE 3840
3830 GOTO 2430
3840 GOTO 2360
3850 IF X5<>77 THEN 3690
3860 VALUE=3.00
3870 CALL SOUND(-500,220,1)
3880 CALL SOUND(-500,277,1)
3890 CALL SOUND(-500,330,1)
3900 CALL SOUND(-740,440,0,659,1,1109,1)
3910 SON=SON-1
3920 CALL SOUND(500,-5,1)
3930 GOTO 3150
```

Fallschirmspringer

Das Spiel läuft auf einem TI99/4A + Extended Basic. Zum Eintippen: Großbuchstaben sind als Großbuchstaben, Kleinbuchstaben als Kleinbuchstaben einzutippen. Die unterstrichenen Buchstaben bedeuten die jeweilige Taste in Verbindung mit der CTRL-Taste. Alle Rems und !s können weggelassen werden.
Nach dem Programmstart erklärt sich das Spiel von selbst.

```
100 ! FALLSCHIRMSPRINGER
110 ! BY STEPHAN ENDRASS
120 ! 7638 BYTES
130 !
140 ! ZEICHENDATEN
150 !
160 ! 33-41
170 !
180 DATA 1818181818ØØ1818,6C6C6C,6CFEFE6C6CFEFE6C,6CFEDØFC7E16FE
    6C,C6CE1E3C78FØE6C6,7ØF8DØ767ECCFE76,18182,18183Ø3Ø3Ø3Ø1818,
    3Ø3Ø1818181833Ø3
190 !
200 ! 42-49
210 !
220 DATA ØØØØØØ1818,ØØ18187E7E1818,ØØØØØØØØØØØ18182,ØØØØØØ7E7E,ØØ
    ØØØØØØØØØØØ1818,Ø6ØE1E3C78FØEØC,7CFECEDEF6E6FE7C,183878781818
    1818
230 !
240 ! 5Ø-56
250 !
260 DATA 7CFEØE3C78EØFEFE,7CFEØE3C3CØEFE7C,1C3C7CECFEFEØCØC,FEFE
    CØFCFEØ6FE7C,7CFECØFCFEC6FE7C,FEFEØ6ØE1C383Ø3,7CFEC67C7CC6FE
    7C
270 !
280 ! 57-64
290 !
300 DATA 7CFEC6FE7EØ6FE7C,ØØ1818ØØØØ1818,ØØ1818ØØØØ18182,Ø61E7CF
    ØFØ7C1EØ6,ØØ7C7CØØØØ7C7C,CØFØ7C1E1E7CFØC,7CFEØ61E1CØØ1818,7C
    FEC6DEDEDEDEC
310 !
320 ! 65-72
330 !
340 DATA 387CEEC6FEFEC6C6,FCFEC6FCFCC6FEFC,7CFECØCØCØCØFE7C,FCFE
    C6C6C6C6FEFC,FEFECØF8F8CØFEFE,FEFECØF8F8CØCØCØ,7CFECØCECEC6F
    E7C,C6C6C6FEFEC6C6C6
350 !
360 ! 73-8Ø
370 !
380 DATA 1818181818181818,FEFEØ6Ø6Ø6Ø6FE7C,C6CEDCF8F8DCCEC6,CØCØ
    CØCØCØCØFEFE,C6EEFEFED6C6C6C6,C6E6E6F6DECECEC6,7CFEC6C6C6C6F
    E7C,FCFEC6FEFCCØCØCØ
390 !
400 ! 81-88
410 !
420 DATA 7CFEC6C6D6CCFE76,FCFEC6FEFCDCCEC6,7CFECØFC7EØ6FE7C,FEFE
    181818181818,C6C6C6C6C6C6FE7C,C6C6EE6C7C38381,C6C6C6C6D6FEFE
    6C,C6EE7C38387CEEC6
430 !
440 ! 89-1Ø4
450 !
460 DATA C6CEFC783Ø3Ø3Ø3,FEFEØE1C387ØFEFE,,8ØCØEØ7F7F,,ØØØØ7CFFF
    E,,FFFFFF,,FFFFFF,,,ØØØ5Ø5Ø7Ø1Ø1Ø3Ø6,,ØØAØAØEØ8Ø8ØCØ6,ØF7Ø88
    442412ØBØ4
470 !
480 ! 1Ø5-115
490 !
500 DATA Ø5Ø5Ø3Ø3Ø3Ø1Ø1Ø3,FØØE11222448DØ2,AØAØCØCØCØ8Ø8ØC,,,,FFF
    FFFFFFFFFFFFF,Ø1Ø1Ø1Ø1Ø1Ø1Ø1Ø1,Ø3Ø7ØF1F1FØFØ7Ø3,Ø3ØF3FFFFF3F
    ØFØ3,CØFØFCFFFFFCFØCØ
510 !
520 ! 116-123
530 !
540 DATA CØEØFØF8F8FØEØCØ,8Ø8Ø8Ø8Ø8Ø8Ø8Ø8Ø,,,FFFFFFFFFFFFFFFF,ØØ
    ØØØØØØØ1Ø3Ø37B,ØØØ37FFFFFFFFFFF,FØF8FCFCFCFCF8FC
550 !
560 ! 124-13Ø
```

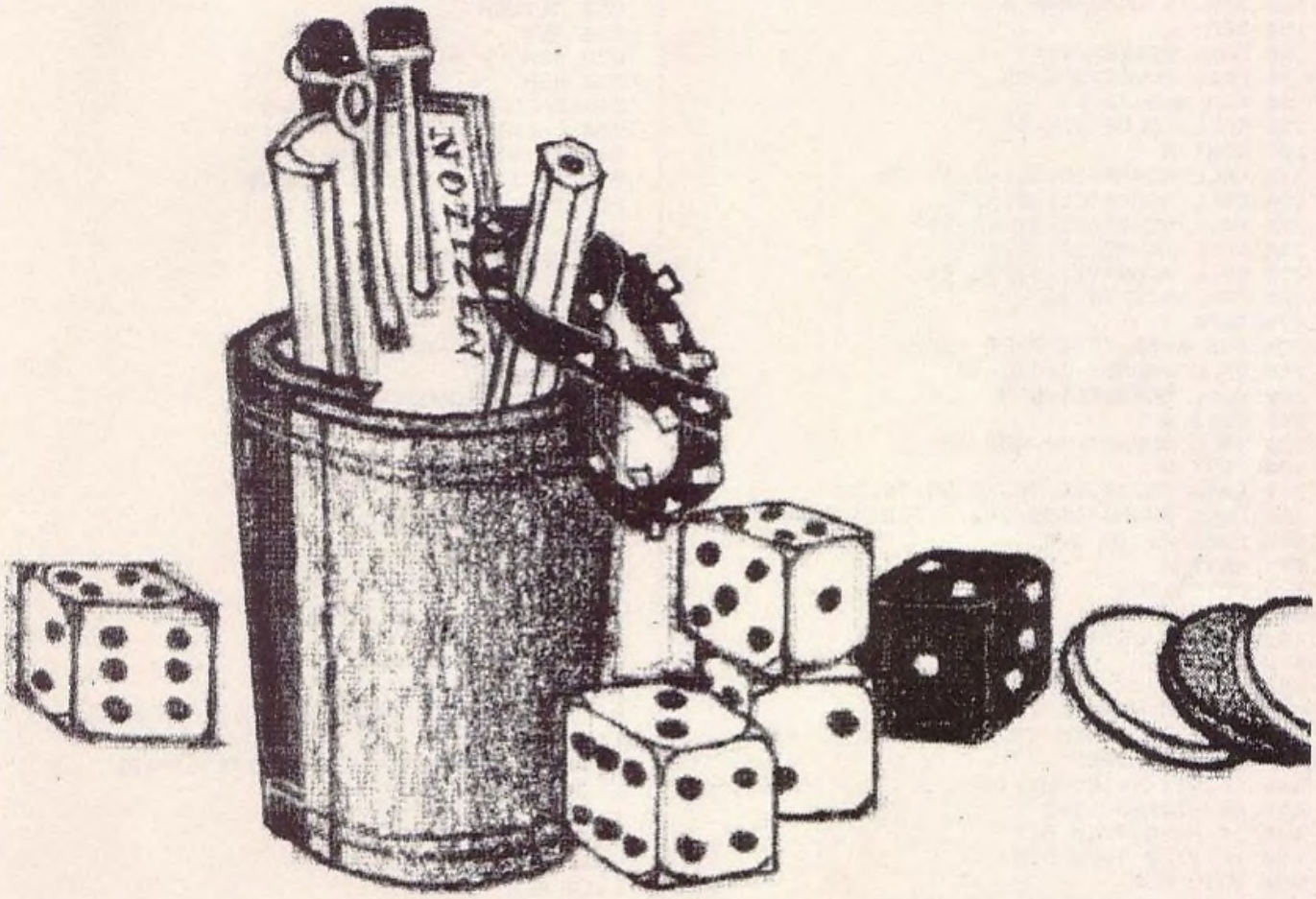
```
570 !
580 DATA ØØØ3Ø7ØFØFØ7ØB3F,FEFFFFFFFFFFFFFF,ØØØØØØ8Ø8Ø8ØØØØF8,7FFF
    FFFFFF7F7F3F,FCFEFFFFFFFFEF8,ØØØØØØØØØØØ1Ø33B,ØØØØØØ31FBFFF
    FFF
590 !
600 ! 131-137
610 !
620 DATA ØØ78FCFEFEFEFEFE,7FFFFFFFFF7F3FØF,BØF8FCFEFFFFFEFC,ØØØ1
    Ø3Ø3Ø31F7FFF,7DFFFFFFFFFFFFFF,CØEØF8F83ØFCFFFF,ØØØØØØØ1Ø1Ø3Ø7
    Ø7
630 !
640 ! 138-143
650 !
660 DATA FEFCFEFEFEFEFCFC,Ø1ØØØØØ1Ø3Ø7ØFØ7,F8CØEØF8FCFEFEFE,Ø3ØF
    ØF1F3F7F7F1F,FFFFFFFFFFFFFFFC,FEFCF8CØ8Ø8
670 DATA ru,qu,pu,pt,ps
680 !
690 ! INITIALISIERUNG
700 !
710 DIM M(2Ø),N(4),SUMME(4)
720 M(19)=-4 :: M(6)=4
730  FOR A=Ø TO 4 :: N(A)=(A-2)*3 :: NEXT A
740 RANDOMIZE
750 CALL CLEAR :: CALL MAGNIFY(3) :: CALL SCREEN(6)
760 FOR A=33 TO 143 :: READ B$ :: CALL CHAR(A,B$):: NEXT A
770 FOR A=Ø TO 4 :: READ A$(A):: NEXT A
780 FOR A=1 TO 8 :: CALL COLOR(A,16,1):: NEXT A
790 CALL COLOR(1Ø,4,1,11,11,1,12,16,1,13,16,1,14,16,1)
800 CALL EINLEITUNG(SPIELER)
810 FOR A=1 TO 4 :: SUMME(A)=Ø :: NEXT A
820 !
830 ! BILD
840 !
850 CALL HCHAR(22,1,111,96)
860 DISPLAY AT(23,1)SIZE(5):"ZEIT:"
870 DISPLAY AT(23,18)SIZE(7):"PUNKTE:"
880 DISPLAY AT(1,1):"SPIELER:"
890 !
900 ! RUNDENSCHLEIFE
910 !
920 FOR RUNDE=1 TO 3
930 CALL HCHAR(2,1,32,64Ø)
940 CALL PUNKTE(Ø):: CALL ZEIT(Ø)
950 CALL SPRITE(§1Ø,96,12,169,121)
960 !
970 ! WOLKEN
980 !
990 J=INT(RND*23)+1 :: H=INT(RND*5)+2
1000 DISPLAY AT(H,J)SIZE(4):" yz{" :: DISPLAY AT(H+1)SIZE(5):"¦xx
     }~" :: DISPLAY AT(H+2,J)SIZE(5):CHR$(127);"xxx‗"
1010 Y=INT(RND*23)+1 :: IF Y>J-5 AND Y<J+5 THEN 1Ø1Ø ELSE H=INT(R
     ND*6)+2
1020 DISPLAY AT(H,Y)SIZE(3):"ABC" :: DISPLAY AT(H+1,Y)SIZE(4):"Dx
     xE"
1030 Z=INT(RND*23)+1 :: IF (Z>J-5 AND Z<J+5)OR(Z>Y-4 AND Z<Y+4)TH
     EN 1Ø3Ø ELSE H=INT(RND*5)+1
1040 DISPLAY AT(H,Z)SIZE(4):" FGH" :: DISPLAY AT(H+1,Z)SIZE(4):"I
     xxJ" :: DISPLAY AT(H+2,Z)SIZE(4):"KxxL" :: DISPLAY AT(H+3,Z)
     SIZE(4):"MxNO"
1050 DISPLAY AT(2Ø,6):"pu" :: DISPLAY AT(21,6):"pu"
1060 !
1070 ! SPIELERSCHLEIFE
1080 !
1090 FOR ANZAHL=1 TO SPIELER :: S=SUMME(ANZAHL):: T=Ø :: U=Ø
1100 CALL VCHAR(1,12,ASC(STR$(ANZAHL)))
1110 CALL PUNKTE(S):: CALL ZEIT(Ø)
1120 FOR PAUSE=1 TO 8ØØ :: NEXT PAUSE
1130 !
1140 ! START FLUGZEUG
1150 !
1160 CALL SPRITE(§5,92,2,4,1,Ø,15):: R=Ø :: CALL SOUND(1ØØ,11Ø,5,
     -5,5)
1170 FOR VERZOEGERUNG=1 TO 1ØØ :: NEXT VERZOEGERUNG
1180 CALL MOTION(§1Ø,Ø,(RUNDE-3)*3)
1190 CALL KEY(1,B,C):: IF C=1 THEN 124Ø
1200 T=T+.1 :: CALL ZEIT(T):: GOTO 119Ø
1210 !
1220 ! ABSPRUNG
1230 !
1240 CALL POSITION(§5,B,C):: CALL SPRITE(§4,1ØØ,11,B+6,C,4,Ø)
1250 T=T+.7
1260 CALL KEY(1,B,C):: IF C=1 THEN 131Ø ELSE T=T+.2 :: R=R+7 :: C
     ALL MOTION(§4,R+8,Ø):: CALL ZEIT(T)
1270 CALL POSITION(§4,B,C):: IF B<11Ø THEN 126Ø ELSE 152Ø
```

```
1280 !
1290 ! FALLSCHIRM OEFFNEN
1300 !
1310 CALL SOUND(-1000,2000,5,2200,6,2400,5,-6,4)
1320 CALL MOTION(§4,4,0):: CALL PATTERN(§4,1004):: T=T+1
1330 CALL KEY(1,A,B):: CALL MOTION(§4,4,M(A+1)+U):: T=T+.2 :: IF
     INT(RND*4)=1 THEN U=N(INT(RND*5)):: DISPLAY AT(20,6):A$(U/3+
     2)
1340 DISPLAY AT(23,6)SIZE(5):USING"§§§.§":T :: CALL POSITION(§4,F
     ,G):: IF F<151 THEN 1330
1350 !
1360 ! AUSWERTUNG
1370 !
1380 CALL MOTION(§4,0,0,§10,0,0):: CALL POSITION(§4,B,C,§10,D,E):
     : IF C<E-9 OR C>E+9 THEN 1460
1390 !
1400 ! LANDUNG GEGLUECKT
1410 !
1420  CALL SOUND(-100,-1,0):: S=S+200 :: CALL PUNKTE(S):: CALL DE
      LSPRITE(§4):: T=T-1 :: IF T>99 THEN 1550 ELSE 1160
1430 !
1440 ! FALSCHE LANDUNG
1450 !
1460 T=T+3 :: CALL SOUND(-10,110,5,-5,5):: CALL ZEIT(T):: IF T>99
     THEN 1550 :: S=S-50 :: CALL DELSPRITE(§4):: IF S<0 THEN S=0
1470 CALL PUNKTE(S):: GOTO 1160
1480 !
1490 ! FALLSCHIRM NICHT
1500 ! GEOEFFNET
1510 !
1520 CALL POSITION(§4,B,C):: IF B<145 THEN 1520
1530 CALL MOTION(§4,0,0):: T=T+6 :: CALL ZEIT(T):: CALL SOUND(99,
     -2,0):: S=S-100 :: IF S<0 THEN S=0
1540 CALL PUNKTE(S):: CALL DELSPRITE(§4):: IF T<99 THEN 1160
1550 IF P>1 THEN CALL SOUND(50,500,3)
1560 !
1570 ! NEUE RUNDE
1580 !
1590 CALL DELSPRITE(§4):: SUMME(ANZAHL)=S :: NEXT ANZAHL :: NEXT
     RUNDE :: GOSUB 1690
1600 !
1610 ! ENDE
1620 !
1630 CALL KEY(0,B,C):: IF B=87 THEN 800
1640 IF B=69 THEN END ELSE CALL ZEIT(999.9):: CALL PUNKTE(9999)::
      DISPLAY AT(1,16)SIZE(9):"" :: CALL ZEIT(T):: CALL PUNKTE(S)
1650 DISPLAY AT(1,16)SIZE(9):"GAME OVER" :: GOTO 1630
1660 !
1670 ! UPRO VERABSCHIEDUNG
1680 !
1690 FOR A=1 TO SPIELER :: DISPLAY AT(9+2*A,1):"SPIELER";A;SUMME(
     A):: NEXT A
1700 DISPLAY AT(1,16)SIZE(9):"GAME OVER"
1710 DISPLAY AT(13,18)SIZE(10):"WEITER MIT"
1720 DISPLAY AT(15,18)SIZE(7):"<< W >>"
1730 DISPLAY AT(17,18)SIZE(8):"ENDE MIT"
1740 DISPLAY AT(19,18)SIZE(7):"<< E >>" :: RETURN
1750 !
1760 ! UPRO SPIELEREINGABE
1770 !
1780 SUB EINLEITUNG(S)
1790 CALL CLEAR :: CALL DELSPRITE(ALL):: DISPLAY AT(3,6):"FALLSCH
     IRMSPRINGER"
1800 DISPLAY AT(8,4):"WIEVIELE SPIELER SIND"
1810 DISPLAY AT(10,4):"SIE ? (EINGABE 1-4)"
1820 DISPLAY AT(12,4):"SIE KOENNEN MIT << W >>",,"   ABSPRINGEN U
     ND IHREN",,"   FALLSCHIRM OEFFNEN.",,"   NACH DEM OEFFNEN GI
     LT:"
1830 DISPLAY AT(22,6):"<< Q >>  NACH LINKS",,,"<< E >> NACH RECHT
     S"
1840 CALL SPRITE(§1,104,16,1,1,4,3,§2,104,2,1,256,4,-3)
1850 CALL KEY(0,A,B):: IF A< 49 XOR A>52 THEN 1850
1860 S=A-48 ::CALL CLEAR :: CALL DELSPRITE(ALL)
1870 SUBEND
1880 !
1890 ! PUNKTE AUSDRUCKEN
1900 !
1910 SUB PUNKTE(S):: DISPLAY AT(23,25)SIZE(4):USING "§§§§":S  ::
     SUBEND
1920 !
1930 ! ZEIT AUSDRUCKEN
1940 !
1950 SUB ZEIT(T):: DISPLAY AT(23,6)SIZE(5):USING "§§§.§":T :: SUB
     END
```

Kniffel

für den TI-99/4A

"Kniffel" wurde schon von den Landsern im Ersten Weltkrieg, unter dem Namen "Hindenburgspiel", geschätzt. Sehr bekannt gewesen ist es vor allem in den zwanziger- und dreißiger Jahren in Skandinavien unter dem Namen "Yahtzee".



Nachdem Anzahl und Namen der Spieler eingegeben worden sind, ruft der Computer automatisch für jeden Spieldurchgang die Namen auf.
Jeder Spieler hat max. 3 Würfe mit bis zu 5 Würfeln pro Durchgang und muß versuchen, eine für ihn sinnvolle Zahlenkombination zu erreichen.
Mit den Tasten 1 - 5 lassen sich entsprechend die jeweiligen Zahlen des Wurfes (Taste 3=3. Zahl des Wurfes) die gesammelt werden wollen, in das nächste Wurffeld schreiben.
Mit der Leertaste würfelt man die noch übriggebliebenen, nicht gesammelten Würfel dazu.
Durch Drücken der Taste "R" läßt sich ein Vertippen der Tasten "1 - 5" wieder rückgängig machen.
Hat man alle 3 Würfe verbraucht oder schon vorher eine geeignete Zahlenkombination erreicht, kann nun die Punktzahl für diese Zahlenreihe durch Drücken der davorstehenden Taste eingetragen und gespeichert werden, wobei die gewürfelten Zahlen für die Kombination G-L noch geordnet werden müssen. Hierzu muß die Taste "Q" gedrückt werden und die Zahlen wie folgt geordnet (mit Hilfe der Tasten 1 5; s.o.):

G-3er-Pasch:	55514	xxxyz
H-4er-Pasch:	333336	xxxxy
I Full House:	66622	xxxyy
K-Straße (kl.):	23453	uvwxw
L-Straße (gr.):	12345	vwxyz

Falls die gewurfelte Zahlenkombination nicht in die Liste eingetragen werden kann, muß ein beliebiges, noch freies Zahlenfeld der Liste gestrichen werden (d.h. Punktzahl = O). Dafür braucht nur die Taste "S" gedrückt werden, und anschließend die Taste des zu streichenden Zahlenfeldes.
Nachdem der Spieler seine Zahlenkombination eingetragen hat, muß er die ENTER-Taste drücken, und der Computer ruft den nächsten Spieler auf.
Wenn alle 13 Durchgänge vorüber sind, zeigt der Computer automatisch die Punkte aller Spiele an und erstellt die Siegertabelle.
Falls ein Spieler zum zweitenmal einen KNIFFEL gewürfelt hat, muß er diesen und alle folgenden in der Zeile O-Bonus eintragen, wofür er je 100 Punkte bekommt. Dieser Bonus ist nur ein Zusatz und zählt nicht als eigener Durchgang.

```
10 REM  ************
20 REM  *          *
30 REM  * KNIFFEL *
40 REM  *          *
50 REM  ************
60 REM
100 CALL CLEAR
110 OPTION BASE 1
120 DIM X(5),D(14,9),NAME$(5),SUM(9)
130 REM
140 REM  * VORSPANN *
150 REM
160 CALL SCREEN(16)
170 CALL COLOR(2,9,1)
180 FOR N=5 TO 12
190 CALL COLOR(N,N,1)
200 NEXT N
210 CALL HCHAR(10,11,42,9)
220 CALL HCHAR(11,19,42)
230 CALL HCHAR(12,11,42,9)
240 CALL HCHAR(11,11,42)
250 CALL HCHAR(11,12,88,7)
260 FOR H=12 TO 18
270 READ Z
280 FOR N=90 TO Z STEP -1
290 CALL SOUND(-1,333,3)
300 CALL HCHAR(11,H,N)
310 NEXT N
320 CALL SOUND(100,300,3)
330 NEXT H
340 DATA 75,78,73,70,70,69,76,32
350 CALL SOUND(1500,262,3,330,3,392,3)
360 FOR N=1 TO 500
370 NEXT N
380 GOTO 1400
390 REM
400 REM  * UNTERPROGRAMM *
410 REM
420 IF KEY-18 THEN 450
430 CALL GCHAR(KEY+3,20,Z)
440 IF Z<>32 THEN 2520
450 H=INT(Y/100)
460 Z=INT((Y-100*H)/10)
470 E=Y-100*H-10*Z
480 IF H<>0 THEN 510
490 IF Z<>0 THEN 520
500 GOTO 530
510 CALL HCHAR(KEY+3,18,H+48)
520 CALL HCHAR(KEY+3,19,Z+48)
530 CALL HCHAR(KEY+3,20,E+48)
540 RETURN
550 Y=X(1)+X(2)+X(3)+X(4)+X(5)
560 FOR N=1 TO 5
570 IF X(N)<>X(N+1)-M THEN 2520
580 NEXT N
590 Y=Y+.01
600 GOSUB 420
610 RETURN
620 Y=0
630 FOR N=1 TO 5
640 IF X(N)<>KEY THEN 660
650 Y=Y+KEY
660 NEXT N
670 Y=Y+.01
680 GOSUB 420
690 RETURN
700 REM
710 REM  * ↑A-F← *
720 REM
730 CALL SOUND(-900,300,2)
740 GOSUB 620
750 D(KEY,A)=Y
760 RETURN
770 REM
780 REM  * ↑G,H← *
790 REM
800 CALL SOUND(-900,300,2)
810 C=KEY-5
820 KEY=KEY+4
830 GOSUB 550
840 D(KEY-4,A)=Y
850 RETURN
860 REM
870 REM  * ↑I,M← *
880 REM
890 CALL SOUND(-900,300,2)
900 C=INT(KEY/3.25)
910 KEY=INT(KEY/C)+KEY
920 IF X(4)<>X(5)THEN 2520
930 IF X(1)=0 THEN 950
940 Y=12.5*C
950 GOSUB 560
960 D(KEY-4,A)=Y
970 RETURN
980 REM
990 REM  * ↑K,L← *
1000 REM
1010 CALL SOUND(-900,300,2)
1020 KEY=KEY+3
1030 C=KEY-11
1040 IF X(1)=0 THEN 1070
1050 M=1
1060 Y=(KEY-11)*10
1070 GOSUB 560
1080 M=0
1090 D(KEY-4,A)=Y
1100 RETURN
1110 REM
1120 REM  * ↑N← *
1130 REM
1140 CALL SOUND(-900,300,2)
1150 KEY=17
1160 Y=X(1)+X(2)+X(3)+X(4)+X(5)+.01
1170 GOSUB 420
1180 D(13,A)=Y
1190 RETURN
1200 REM
1210 REM  * ↑O← *
1220 REM
1230 CALL SOUND(-900,210,2)
1240 IF X(1)=0 THEN 2520
1250 CALL GCHAR(19,19,Z)
1260 IF Z=32 THEN 2520
1270 KEY=18
1280 C=4
10 CALL GCHAR(21,18,Z)
1300 IF Z=32 THEN 1330
1310 Y=(VAL(CHR$(Z))+1)*100
1320 GO
1330 Y=11 GOSUB 560
1350 D(1)=Y
1360 RR1370 REM
1380 REM  * SPIELREGELN *
1390 REM
1400 CALL CLEAR
1410 CALL SCREEN(5)
1420 FOR N=2 TO 8
1430 CALL COLOR(N,16,1)
1440 NEXT N
1450 CALL COLOR(9,2,1)
1460 CALL COLOR(10,2,1)
```

```
1470 PRINT "        ** KNIFFEL **"::"          SPIELREGELN"::
1480 PRINT "EINER      -  NUR 1ER ZAEHLEN":"   ":"SECHSER   -  NUR 6ER ZAEHLEN"
::
1490 PRINT "3ER-PASCH  -  MIND.3 GLEICHE                ZAHLEN":"4ER-PASCH  -  MIN
D.4 GLEICHE              ZAHLEN"
1500 PRINT :"FULL HOUSE  3 GLEICHE                    +2 GLEICHE ZAHLEN"::
1510 PRINT "STRASSE+KL+   -  4ER-FOLGE":"STRASSE+GR+  -  5ER-FOLGE"::
1520 PRINT "KNIFFEL -  5 GLEICHE ZAHLEN'::"CHANCE  - ALLE AUGEN ZAEHLEN"::
1530 CALL HCHAR(24,29,62,2)
1540 CALL KEY(0,KEY,STAT)
1550 IF STAT=0 THEN 1540
1560 CALL CLEAR
1570 PRINT "OBERE HAELFTE: BONUS +35PKT+                WENN GESAMT
  )=63 PKT"::::::
1580 PRINT "UNTERE HAELFTE: BONUS +100+                 JE ZUSAETZ-
     LICHEN":TAB(17);"KNIFFEL"
1590 PRINT ::::::::"                              ))"
1600 CALL KEY(0,KEY,STAT)
1610 IF STAT=0 THEN 1600
1620 CALL CLEAR
1630 PRINT "      TASTENBEDEUTUNG"::::"TASTE       BEDEUTUNG":::" 1-5   SAMMELN DE
R ZAHLEN":::
1640 PRINT "  Q     ORDNEN DER ZAHLEN":::"  S    STREICHEN VON ZAHLEN  ":::
1650 PRINT "  R     VERBESSERN VOM              VERTIPPEN"::
1660 CALL KEY(0,KEY,STAT)
1670 IF STAT=0 THEN 1660
1680 REM
1690 REM  * SPIELSTART *
1700 REM
1710 CALL CLEAR
1720 INPUT "ANZAHL DER SPIELER: ":SPIEL
1730 IF (SPIEL<1)+(SPIEL>9)<0 THEN 1720
1740 FOR A=1 TO SPIEL
1750 INPUT "NAME: ":NAME$(A)
1760 IF LEN(NAME$(A))<20 THEN 1000
1770 CALL SOUND(200,220,3)
1780 PRINT :' * NAME TOO LONG *"::
1790 GOTO 1750
1800 NEXT A
1810 CALL CLEAR
1820 CALL CHAR(104,"0000000000000040")
1830 CALL CHAR(105,"000000000000FF")
1840 CALL CHAR(106,"0000000000000102")
1850 CALL CHAR(107,"0202020202020202")
1860 CALL CHAR(108,"0201")
1870 CALL CHAR(109,"00FF")
1880 CALL CHAR(110,"4080")
1890 CALL CHAR(111,"4040404040404040")
1900 CALL CHAR(100,"0020100804081020")
1910 PRINT "   NAME:":"    NR. :":"                  PKT"
1920 PRINT " A-EINER                1. WURF":" B-ZWEIER":" C-DREIER":" D-VIERER":"
E-FUENFER":" F-SECHSER"
1930 PRINT "     GESAMT":"    (BONUS)              2. WURF":"    GESAMT+1+":::
1940 PRINT " G-3ER-PASCH":" H-4ER-PASCH":" I-FULL HOUSE":" K-STRASSE+KL+":' L-ST
RASSE+GR+        3. WURF"
1950 PRINT " M-KNIFFEL":" N-CHANCE":" O-BONUS":"    GESAMT+2+"::
1960 FOR N=1 TO 8
1970 CALL HCHAR(24,N+5,ABS(SEG$("ENDSJMNE",N,1)))
1980 NEXT N
1990 CALL HCHAR(10,5,100)
2000 CALL HCHAR(12,5,100)
2010 CALL HCHAR(22,5,100)
2020 CALL HCHAR(10,16,100)
2030 CALL HCHAR(12,15,100,2)
2040 CALL HCHAR(22,15,100,2)
2050 CALL HCHAR(24,5,100)
2060 CALL VCHAR(4,17,100,9)
2070 CALL VCHAR(14,17,100,9)
2080 CALL HCHAR(24,14,100,4)
2090 FOR N=7 TO 21 STEP 7
2100 CALL HCHAR(N-1,25,106)
2110 CALL HCHAR(N,25,107)
2120 CALL HCHAR(N+1,25,108)
2130 CALL HCHAR(N-1,26,105,5)
2140 CALL HCHAR(N+1,26,109,5)
2150 CALL HCHAR(N-1,31,104)
2160 CALL HCHAR(N,31,111)
2170 CALL HCHAR(N+1,31,110)
2180 NEXT N
2190 A=0
2200 A=A+1
2210 IF A>SPIEL THEN 2190
2220 CALL HCHAR(1,12,32,20)
2230 CALL HCHAR(2,12,32,2)
2240 CALL VCHAR(4,18,32,21)
2250 CALL VCHAR(4,19,32,21)
```

```
2260 CALL VCHAR(4,20,32,21)
2270 FOR N=1 TO LEN(NAME$(A))
2280 CALL HCHAR(1,N-11,ASC(SEG$(NAME$(A),N,1)))
2290 NEXT N
2300 CALL HCHAR(2,12,A+48)
2310 FOR N=1 TO 13
2320 IF D(N,A)=0 THEN 2360
2330 Y=D(N,A)
2340 KEY=N+INT(N/7)*4
2350 GOSUB 450
2360 NEXT N
2370 IF D(14,A)=0 THEN 2410
2380 Y=D(14,A)
2390 KEY=18
2400 GOSUB 450
2410 IF ANZ=13*SPIEL THEN 3330
2420 R=0
2430 I=0
2440 RANDOMIZE
2450 R=R+7
2460 FOR N=I+1 TO 5
2470 X(N)=INT(6*RND)+1
2480 CALL HCHAR(R,N+25,X(N)+48)
2490 NEXT N
2500 I=0
2510 W$=""
2520 M=0
2530 CALL KEY(0,KEY,STAT)
2540 IF STAT=0 THEN 2530
2550 IF STAT=-1 THEN 2530
2560 IF KEY=82 THEN 3160
2570 IF R=14.1 THEN 2810
2580 IF KEY>53 THEN 2760
2590 IF R=21 THEN 2520
2600 IF KEY=32 THEN 2440
2610 IF KEY<49 THEN 2520
2620 IF KEY>53 THEN 2520
2630 KEY=KEY-48
2640 IF I=5 THEN 2530
2650 FOR N=1 TO I
2660 IF KEY=VAL(SEG$(W$,N,1))THEN 2530
2670 NEXT N
2680 W$=W$&STR$(KEY)
2690 I=I+1
2700 CALL GCHAR(R,KEY+25,Z)
2710 X(I)=VAL(CHR$(Z))
2720 CALL HCHAR(R+7,I+25,X(I)+48)
2730 IF I<5 THEN 2750
2740 R=21
2750 GOTO 2530
2760 IF KEY<65 THEN 2530
2770 IF KEY>83 THEN 2530
2780 IF R=21 THEN 2820
2790 IF I=0 THEN 2820
2800 CALL GCHAR(R+7,30,Z)
2810 IF Z=32 THEN 2530
2820 I=0
2830 W$=""
2840 Y=0
2850 KEY=KEY-64
2860 ON KEY GOSUB 710,710,710,710,710,710,780,780,870,2530,990,990,870,1120,1210
,2530,2990,2530,3080
2870 IF KEY=18 THEN 2900
2880 CALL KEY(0,KEY,STAT)
2890 IF KEY<>13 THEN 2880
2900 CALL HCHAR(7,25,32,5)
2910 CALL HCHAR(14,25,32,5)
2920 CALL HCHAR(21,25,32,5)
2930 IF KEY=18 THEN 2720
2940 ANZ=ANZ+1
2950 IF ANZ=SPIEL*13 THEN 3290
2960 IF SPIEL=1 THEN 2420
2970 GOTO 2200
2980 REM
2990 REM  * ORDNEN *
3000 REM
3010 FOR N=1 TO 5
3020 CALL HCHAR(21,25-N,32)
3030 CALL HCHAR(14,25-N,X(N)+48)
3040 NEXT N
3050 R=14.1
3060 GOTO 2530
3070 REM
3080 REM  * STREICHEN *
3090 REM
3100 CALL SOUND(-300,500,3)
```

```
3110 FOR N=1 TO 5
3120 X(N)=0
3130 NEXT N
3140 GOTO 2530
3150 REM
3160 REM  * VERBESSERN *
3170 REM
3180 CALL SOUND(300,200,3)
3190 IF R=21 THEN 2530
3200 CALL GCHAR(R+7,30,Z)
3210 IF Z<>32 THEN 2530
3220 FOR N=1 TO E
3230 CALL GCHAR(R,N+25,Z)
3240 X(N)=VAL(CHR$(Z))
3250 CALL HCHAR(R+7,N+25,32)
3260 NEXT N
3270 GOTO 2500
3280 REM
3290 REM  * AUSRECHNEN *
3300 REM
3310 FOR A=1 TO SPIEL
3320 GOTO 2220
3330 Y=INT(D(1,A)+D(2,A)+D(3,A)+D(4,A)+D(5,A)+D(6,A))
3340 KEY=7
3350 GOSUB 450
3360 IF Y<63 THEN 3400
3370 Y=Y+35
3380 CALL HCHAR(11,19,51)
3390 CALL HCHAR(11,20,53)
3400 KEY=9
3410 GOSUB 450
3420 SUM(A)=Y
3430 Y=INT(D(7,A)+D(8,A)+D(9,A)+D(10,A)+D(11,A)+D(12,A)+D(13,A)+D(14,A))
3440 KEY=19
3450 GOSUB 450
3460 SUM(A)=SUM(A)+Y
3470 KEY=21
3480 Y=SUM(A)
3490 GOSUB 450
3500 REM
3510 REM  * SIEGERTABELLE *
3520 REM
3530 CALL KEY(0,KEY,STAT)
3540 IF STAT=0 THEN 3530
3550 NEXT A
3560 CALL CLEAR
3570 CALL SCREEN(7)
3580 Z=0
3590 FOR N=1 TO 20
3600 CALL SOUND(-200,-1,3)
3610 CALL SOUND(-200,-2,3)
3620 NEXT N
3630 PRINT TAB(5);"** SIEGERTABELLE **"
3640 FOR N=1 TO INT(22/(SPIEL+1))-1
3650 PRINT
3660 NEXT N
3670 FOR Z=1 TO SPIEL
3680 FOR H=Z+1 TO SPIEL
3690 IF SUM(H)<SUM(Z)THEN 3760
3700 E=SUM(Z)
3710 SUM(Z)=SUM(H)
3720 SUM(H)=E
3730 W$=NAME$(Z)
3740 NAME$(Z)=NAME$(H)
3750 NAME$(H)=W$
3760 NEXT H
3770 PRINT TAB(SGN(10-Z)+1);STR$(Z);". ";NAME$(Z);TAB(25);SUM(Z)
3780 FOR N=1 TO INT(22/(SPIEL+1))-1
3790 PRINT
3800 NEXT N
3810 NEXT Z
3820 CALL KEY(0,KEY,STAT)
3830 IF STAT=0 THEN 3820
3840 CALL CLEAR
3850 ANZ=0
3860 FOR N=1 TO SPIEL
3870 SUM(N)=0
3880 NAME$(N)=""
3890 FOR H=1 TO 14
3900 D(H,N)=0
3910 NEXT H
3920 NEXT N
3930 INPUT "NOCH EIN SPIEL?      (J)  (N)                    ":W$
3940 IF ASC(W$)=74 THEN 1710
3950 IF ASC(W$)<>78 THEN 3930
3960 END
```

Mauerklauer

für den TI 99/4

Das Spiel "Mauerklauer" wird mit zwei verschiedenen Figuren gespielt: Mit den "Wächter-Figuren" und der "Eigenen Figur". Auf dem Bildschirm ist eine Mauer aufgebaut, über die Sie klettern müssen um ins Freie zu gelangen.

Dies geht aber nur, wenn Sie die Mauersteine abräumen und somit Punkte sammeln.
Jetzt treten die Wächter in Aktion, verfolgen Sie und versuchen Sie natürlich auch zu fangen.
Auf dem Spielfeldrand befinden sich mehrere unsichtbare Punkte, auf denen die eigene Figur versetzt wird. Der Ort der Versetzung ist zufällig, es gibt für jede Versetzung 50 Punkte. Ihr Ziel ist es, über den Bildschirmrand auf die andere Seite zu entkommen, denn dorthin können Ihnen die Wächter nicht folgen.

```
100 REM  ******************
110 REM  *** MAUERKLAUER ****
120 REM  * PROGRAMMIERT VON *
130 REM  *** H-P SCHWANECK **
140 REM  *** BRAUNSCHWEIG ***
150 REM  ******************
160 CALL CHAR(126,"183C5A7E7E427E5A")
170 CALL CHAR(35,"3C7E7E7E7E7E7E3C")
180 CALL CHAR(42,"1898FF3D3C3C2424")
190 CALL COLOR(12,13,10)
200 CALL COLOR(1,7,10)
210 CALL CLEAR
220 PRINT "****************************"
230 PRINT
240 PRINT "     MAUERKLAUER     "
250 PRINT
260 PRINT "****************************"
270 FOR TE=1 TO 14
280 PRINT
290 NEXT TE
300 GOSUB 3630
310 PRINT "SPIELREGELN?(J-N)"
320 CALL KEY(0,K,S)
330 IF K=74 THEN 360
340 IF K=78 THEN 720
350 GOTO 320
360 CALL CLEAR
370 CALL SCREEN(10)
380 PRINT "EIGENE FIGUR  *"
390 PRINT
400 PRINT "WAECHTER FIGUR   ~ "
410 PRINT
420 PRINT "DIE WAECHTER VERFOLGEN SIE  UND VERSUCHEN SIE ZU FANGEN.DIE WAECHTER
KOENNEN ABER   NICHT UEBER DIE";
430 PRINT "MAUERN KLETTERN"
440 PRINT
450 PRINT "SIE SAMMELN PUNKTE,INDEM SIEDIE MAUERSTEINE ABRAEUMEN."
460 PRINT
470 PRINT "AUF DEM SPIELFELD BEFINDEN  SICH MEHRERE UNSICHTBARE    PUNKTE,AUF DE
NEN DIE EIGENE FIGUR  VESETZT";
480 PRINT " WIRD."
490 PRINT "DER ORT DER VERSETZUNG IST  ZUFAELLIG. ES GIBT FUER     JEDE VERSETZU
NG 50 PUNKTE."
500 PRINT
510 PRINT "WEITER MIT SPACE TASTE."
520 CALL KEY(0,KEY,STATUS)
530 IF KEY<>32 THEN 520
540 CALL CLEAR
550 PRINT "SIE KOENNEN UEBER DEN       BILDSCHIRMRAND AUF DIE      ANDERE SEITE
ENTKOMMEN,"
560 PRINT
```

```
570 PRINT "DENN DIE WAECHTER KOENNEN   DORTHIN NICHT FOLGEN."
580 PRINT
590 PRINT "ES WIRD EIN ZEITLIMIT       GESETZT.SIE MUESSEN SICH    ALSO BEEILEN
UM ALLE MAUERN ZU BESEITIGEN."
600 PRINT
610 PRINT "GESTEUERT WIRD MIT DEN TASTEN:"
620 PRINT
630 PRINT "       K(OBEN)  "
640 PRINT "A(LINKS)    S(RECHTS)"
650 PRINT "       M(UNTEN)"
660 PRINT
670 PRINT "ALLES VERSTANDEN?           DANN BITTE DIE SPACE TASTE  DRUECKEN."
680 CALL KEY(0,KEY,STATUS)
690 IF KEY<>32 THEN 680
700 SCORE=0
710 CALL CLEAR
720 ZE$="DIE ZEIT IST ABGELAUFEN"
730 VE$="SIE SIND GEFANGEN WORDEN    VERSUCHEN SIE ES NOCHMAL"
740 INPUT "WIE VIELE MAUERN?":MAUER
750 IF MAUER<40 THEN 790
760 PRINT
770 PRINT "ZU VIELE MAUERN.VERSUCHEN   SIE ES NOCH EINMAL."
780 GOTO 740
790 IF MAUER>9 THEN 830
800 PRINT
810 PRINT "ZU WENIG MAUERN.VERSUCHEN   SIE ES NOCH EINMAL."
820 GOTO 740
830 INPUT "LEGEN SIE DEN SCHWIERIGKEITSGRAD DURCH DIE ANZAHL DER   GEGNER FEST!
":SGRAD
840 IF SGRAD>1 THEN 880
850 PRINT
860 PRINT "LEIDER ZU WENIG GEGNER.     WAEHLEN SIE NOCH EINMAL."
870 GOTO 830
880 IF SGRAD<5 THEN 920
890 PRINT
900 PRINT "ZU VIELE GEGNER.WAEHLEN SIE NOCH EINMAL."
910 GOTO 830
920 PRINT "ZUM START SPACE TASTE       DRUECKEN UND 2 MIN. WARTEN."
930 CALL KEY(0,KEY,STATUS)
940 IF KEY<>32 THEN 930
950 GOSUB 3630
960 CALL COLOR(12,10,12)
970 CALL COLOR(1,7,12)
980 SCORE=0
990 ZEIT=(9*SGRAD)*MAUER
1000 CALL CLEAR
1010 CALL SCREEN(12)
1020 STEINZAHL=0
1030 FOR M=0 TO MAUER
1040 RANDOMIZE
1050 A=INT(2*RND)
1060 X=INT(22*RND+3)
1070 Y=INT(16*RND+2)
1080 IF A<1 THEN 1110
1090 CALL HCHAR(Y,X,35,7)
1100 GOTO 1120
1110 CALL VCHAR(Y,X,35,7)
1120 NEXT M
1130 REM  VERSETZUNGEN
1140 FOR VERS=1 TO SGRAD*5
1150 VX=INT(30*RND+2)
1160 VY=INT(22*RND+2)
1170 CALL COLOR(9,1,1)
1180 CALL HCHAR(VY,VX,97)
```

```
1190 NEXT VERS
1200 REM  STEINE ZAEHLEN
1210 FOR ZEIL=1 TO 24
1220 FOR SPAL=1 TO 32
1230 CALL GCHAR(ZEIL,SPAL,INHALT)
1240 IF INHALT<>35 THEN 1260
1250 STEINZAHL=STEINZAHL+1
1260 NEXT SPAL
1270 NEXT ZEIL
1280 REM  STARTPOSITIONEN
1290 OPTION BASE 1
1300 RANDOMIZE
1310 DIM XR(4)
1320 DIM YR(4)
1330 FOR I=1 TO SGRAD
1340 XR(I)=INT(26*RND+4)
1350 YR(I)=INT(20*RND+2)
1360 CALL GCHAR(YR(I),XR(I)+1,L1)
1370 CALL GCHAR(YR(I),XR(I)-1,L2)
1380 CALL GCHAR(YR(I)+1,XR(I),L3)
1390 CALL GCHAR(YR(I)-1,XR(I),L4)
1400 CALL GCHAR(YR(I),XR(I),L5)
1410 IF (L1+L2+L3+L4+L5)>160 THEN 1340
1420 NEXT I
1430 XS=INT(15*RND+4)
1440 YS=INT(10*RND+4)
1450 CALL GCHAR(YS,XS,L6)
1460 IF L6=35 THEN 1430
1470 FOR I=1 TO SGRAD
1480 CALL HCHAR(YR(I),XR(I),126)
1490 NEXT I
1500 CALL HCHAR(YS,XS,42)
1510 REM  ORIENTIERUNGSSCHLEIFE
1520 IF SGRAD<>1 THEN 1560
1530 FOR W=1 TO 1530
1540 NEXT W
1550 GOTO 1660
1560 IF SGRAD<>2 THEN 1600
1570 FOR W=1 TO 1230
1580 NEXT W
1590 GOTO 1660
1600 IF SGRAD<>3 THEN 1640
1610 FOR W=1 TO 1030
1620 NEXT W
1630 GOTO 1660
1640 FOR W=1 TO 803
1650 NEXT W
1660 VAR=0
1670 CALL SOUND(-330,880,0)
1680 CALL SOUND(-330,988,0)
1690 CALL SOUND(-330,784,0)
1700 CALL SOUND(-330,392,0)
1710 CALL SOUND(-990,576,0)
1720 REM  SPIELBETRIEB
1730 ZEIT=ZEIT-1
1740 IF ZEIT>0 THEN 1910
1750 IF SCORE<HISCORE THEN 1770
1760 HISCORE=SCORE
1770 CALL CLEAR
1780 IF VAR=1 THEN 1800
1790 PRINT ZE$
1800 PRINT "SCORE";SCORE
1810 PRINT "HIGHSCORE";HISCORE
1820 PRINT "NOCHMAL? DANN TASTE J "
1830 FOR T=1 TO 2000
1840 CALL KEY(0,KEY,STATUS)
1850 IF STATUS=0 THEN 1870
1860 IF KEY=74 THEN 1900 ELSE 1880
1870 NEXT T
1880 PRINT "GAME OVER"
1890 END
1900 GOTO 710
1910 REM  BEWEGUNG SPIELERFIGUR
1920 CALL KEY(0,KEY,STATUS)
1930 IF KEY<>75 THEN 2110
1940 CALL VCHAR(YS,XS,32)
1950 YS=YS-1
1960 IF YS>0 THEN 1980
1970 YS=24
1980 CALL GCHAR(YS,XS,VA)
1990 IF VA<>35 THEN 2020
2000 SCORE=SCORE+10
2010 STEINZAHL=STEINZAHL-1
2020 IF VA<>126 THEN 2070
2030 GOSUB 3400
2040 FOR P=1 TO 50
2050 NEXT P
2060 GOTO 1750
2070 IF VA<>97 THEN 2090
2080 GOSUB 3300
2090 CALL HCHAR(YS,XS,42)
2100 GOTO 2660
2110 IF KEY<>77 THEN 2300
2120 CALL VCHAR(YS,XS,32)
2130 YS=YS+1
2140 IF YS<25 THEN 2160
2150 YS=1
2160 CALL GCHAR(YS,XS,VA)
2170 IF VA<>35 THEN 2200
2180 SCORE=SCORE+10
2190 STEINZAHL=STEINZAHL-1
2200 IF VA<>126 THEN 2260
2210 GOSUB 3400
2220 FOR P=1 TO 50
2230 NEXT P
2240 VAR=1
2250 GOTO 1750
2260 IF VA<>97 THEN 2280
2270 GOSUB 3300
2280 CALL HCHAR(YS,XS,42)
2290 GOTO 2660
2300 IF KEY<>83 THEN 2480
2310 CALL VCHAR(YS,XS,32)
2320 XS=XS+1
2330 IF XS<33 THEN 2350
2340 XS=1
2350 CALL GCHAR(YS,XS,VA)
2360 IF VA<>35 THEN 2390
2370 SCORE=SCORE+10
2380 STEINZAHL=STEINZAHL-1
2390 IF VA<>126 THEN 2440
2400 GOSUB 3400
2410 FOR P=1 TO 50
2420 NEXT P
2430 GOTO 1750
2440 IF VA<>97 THEN 2460
2450 GOSUB 3300
2460 CALL VCHAR(YS,XS,42)
2470 GOTO 2660
2480 IF KEY<>65 THEN 2650
2490 CALL VCHAR(YS,XS,32)
2500 XS=XS-1
2510 IF XS>0 THEN 2530
2520 XS=32
2530 CALL GCHAR(YS,XS,VA)
2540 IF VA<>35 THEN 2570
2550 SCORE=SCORE+10
2560 STEINZAHL=STEINZAHL-1
2570 IF VA<>126 THEN 2620
2580 GOSUB 3400
2590 FOR P=1 TO 32
2600 NEXT P
2610 GOTO 1750
2620 IF VA<>97 THEN 2640
2630 GOSUB 3300
2640 CALL VCHAR(YS,XS,42)
2650 REM  STEUERUNG RECHNERFIGUR
2660 IF STEINZAHL>0 THEN 2690
2670 GOSUB 3920
2680 GOTO 993
2690 FOR I=1 TO SGRAD
2700 FOR SCHRITT=1 TO 2
2710 RANDOMIZE
2720 ZUF=INT(2*RND)
2730 IF ZUF<>0 THEN 3010
2740 IF XR(I)<XS THEN 2880
2750 XR(I)=XR(I)-1
2760 IF XR(I)>0 THEN 2780
2770 XR(I)=1
2780 CALL GCHAR(YR(I),XR(I),F)
2790 IF F<>35 THEN 2820
2800 XR(I)=XR(I)+1
2810 GOTO 3020
2820 IF F<>42 THEN 2850
2830 GOSUB 3400
2840 GOTO 1750
2850 CALL VCHAR(YR(I),XR(I)+1,32)
2860 CALL VCHAR(YR(I),XR(I),126)
2870 GOTO 3270
2880 XR(I)=XR(I)+1
2890 IF XR(I)<33 THEN 2910
2900 XR(I)=32
2910 CALL GCHAR(YR(I),XR(I),F)
2920 IF F<>35 THEN 2950
2930 XR(I)=XR(I)-1
2940 GOTO 3150
2950 IF F<>42 THEN 2980
2960 GOSUB 3400
2970 GOTO 1750
2980 CALL VCHAR(YR(I),XR(I)-1,32)
2990 CALL VCHAR(YR(I),XR(I),126)
3000 GOTO 3270
3010 IF YR(I)<YS THEN 3150
3020 YR(I)=YR(I)-1
3030 IF YR(I)>0 THEN 3050
3040 YR(I)=1
3050 CALL GCHAR(YR(I),XR(I),F)
3060 IF F<>35 THEN 3090
3070 YR(I)=YR(I)+1
3080 GOTO 2880
3090 IF F<>42 THEN 3120
3100 GOSUB 3400
3110 GOTO 1750
3120 CALL VCHAR(YR(I)+1,XR(I),32)
3130 CALL VCHAR(YR(I),XR(I),126)
3140 GOTO 3270
3150 YR(I)=YR(I)+1
3160 IF YR(I)<33 THEN 3180
3170 YR(I)=32
3180 CALL GCHAR(YR(I),XR(I),F)
3190 IF F<>35 THEN 3220
3200 YR(I)=YR(I)-1
3210 GOTO 2750
3220 IF F<>42 THEN 3250
3230 GOSUB 3430
3240 GOTO 1750
3250 CALL VCHAR(YR(I)-1,XR(I),32)
3260 CALL VCHAR(YR(I),XR(I),126)
3270 NEXT SCHRITT
3280 NEXT I
3290 GOTO 1720
3300 REM  VERSETZUNG BERECHNEN
3310 XS=INT(30*RND+2)
3320 YS=INT(22*RND+2)
3330 CALL GCHAR(YS,XS,INHALT)
3340 IF INHALT<>32 THEN 3310
3350 FOR T=1 TO 25
3360 CALL SOUND(-5,110+T*100,0)
3370 NEXT T
3380 SCORE=SCORE+50
3390 RETURN
3400 PRINT VE$
3410 FOR C=1 TO 3
3420 FOR S=1 TO 30
3430 CALL SOUND(-1,1000-20*S,S)
3440 NEXT S
3450 NEXT C
3460 VAR=1
3470 CALL SOUND(250,440,0)
3480 CALL SOUND(150,392,0)
3490 CALL SOUND(150,330,0)
3500 CALL SOUND(150,294,0)
3510 CALL SOUND(150,277,0)
3520 CALL SOUND(250,262,0)
3530 CALL SOUND(350,220,0)
3540 CALL SOUND(200,294,0)
3550 CALL SOUND(200,294,0)
3560 CALL SOUND(200,294,0)
3570 CALL SOUND(200,262,0)
3580 CALL SOUND(200,220,0)
3590 CALL SOUND(250,196,0)
3600 CALL SOUND(150,220,0)
3610 CALL SOUND(150,220,0)
3620 RETURN
3630 CALL SOUND(100,220,0)
3640 FOR PAUSE=1 TO 200
3650 NEXT PAUSE
3660 CALL SOUND(100,247,0)
3670 CALL SOUND(100,262,0)
3680 FOR PAUSE=1 TO 200
3690 NEXT PAUSE
3700 CALL SOUND(100,196,0)
3710 CALL SOUND(100,220,0)
3720 FOR PAUSE=1 TO 50
3730 NEXT PAUSE
3740 CALL SOUND(100,247,0)
3750 CALL SOUND(200,262,0)
3760 CALL SOUND(100,349,0)
3770 CALL SOUND(100,330,0)
3780 FOR PAUSE=1 TO 100
3790 NEXT PAUSE
3800 CALL SOUND(200,220,0)
3810 CALL SOUND(200,262,0)
3820 CALL SOUND(200,440,0)
3830 CALL SOUND(400,430,0)
3840 FOR PAUSE=1 TO 50
3850 NEXT PAUSE
3860 CALL SOUND(100,330,0)
3870 CALL SOUND(100,294,0)
3880 CALL SOUND(100,262,0)
3890 CALL SOUND(100,330,0)
3900 CALL SOUND(100,330,0)
3910 RETURN
3920 CALL CLEAR
```

```
3930 PRINT "GRATULIERE! ALLE STEINE WEG-GESCHAFFT.WARTEN SIE CA 2   MINUTEN,BIS
ICH NEUE MAUERN AUFGEBAUT HABE."
3940 GOSUB 3630
3950 RETURN
```

3-D - Neue Dimensionen

Als vor einigen Monaten in den 3. Programmen einige von Stereofilmen faszinierte Enthusiasten des deutschen Fernsehens produzierte Sendungen dreidimensional über die Bildschirme der Nation flimmerten, setzte auf die zum Erreichen des gewünschten Effektes notwendigen Rot-Grün-Brillen ein regelrechter Boom ein. Sicher haben Sie aus dieser Zeit noch eine solche Brille in irgendeiner Schublade. Holen Sie sie heraus und schauen Sie sich zur Einstimmung auf das was nun folgt, das Titelbild dieses Heftes noch einmal mit Brille an.

Was mit einer Stereo-Kamera möglich ist, sagten sich einige englische Computerfreaks, daß muß auch auf dem heimischen Micro zu verwirklichen sein; und entwickelten kurzerhand die ersten echten 3-D-Spiele auf ihren Computern.

Das Ganze ist schließlich eine äußerst einfache und unkomplizierte Sache. So wie unsere beiden Augen zwei Bilder, die vom Gehirn zu einem perspektivischen Bild verschmolzen werden, sehen, bildet man zwei Ansichten eines Objektes auf dem Bildschirm ab.

Da dieses nur zweidimensional geschehen kann, muß man das Gehirn mit einem Trick überlisten - ihm vorgaukeln, es sähe vom entsprechenden Objekt die gewohnten zwei Abbildungen. Je nachdem in welchem Winkel die Augen auf das Objekt gerichtet sind, kann es sich die entsprechenden Entfernungen errechnen, was zum räumlichen Sehen führt. Die Rot-Blau-Brille ist nach dem heutigen Stand der Technik die einzige Möglichkeit, 3-D-Visionen hervorzurufen. Sie sorgt dafür, daß von zwei im bestimmten Abstand übereinandergedruckter Bilder jeweils eines dem entsprechenden Auge zugeführt wird.

Ein Problem, Perspektive auf dem Computerbildschirm zu bringen, liegt hauptsächlich darin, daß die meisten Computer Farben in verschiedenen Stärken darstellen. Meist erscheint Rot hell und Grün im Gegensatz dazu sehr dunkel. Zum Erreichen des 3-D-Effektes ist es aber notwendig, daß beide Farben ungefähr gleich stark erscheinen. Statt Blau kann man auch Grün verwenden, verbaut sich allerdings den Weg zu weiteren Experimenten bezüglich echten Farb-3-D-Grafiken.

Eine weitere Voraussetzung für ein gutes Gelingen ist, daß der Computer möglichst hochauflösende Farbgrafiken darstellen kann.

Was muß ein Programm enthalten, das mit echten 3-D-Effekten abläuft?

Aus den zahlreichen vorstellbaren Möglichkeiten, wollen wir einige Anregungen zum Selbstprogrammieren geben.

Wie bereits erwähnt, müssen 2 Bilder in Rot und Blau auf dem Bildschirm abgebildet werden, es fragt sich nur, in welchem Abstand voneinander.

Als Anhaltspunkt nehmen wir die Bildschirmebene. Soll das Bild direkt darauf abgebildet sein, müssen beide Augen es an der gleichen Stelle sehen - es wird also ein Bild über das andere gedruckt. Soll das Bild vor der Bildschirm-Ebene erscheinen, dann muß das rote Abbild links vom blauen abgedruckt werden und soll es dahinterliegen, dann ist es umgekehrt. Je weiter die beiden Abbilder auseinandergedruckt werden, um so näher bzw. entfernter erscheint uns das Objekt.

Dieses machen wir uns zunutze, indem wir uns eine Serie errechnen, die wir später in unserem Programm anwenden. 1; 2; 4; 8; 16 wäre so eine Serie, aber die Auflösung aller Homecomputer ist dafür zu gering. 1; 1,25; 1,5625; 1,953125; 2,44140625; usw. (immer mal 1,25) reicht für die meisten Zwecke aus und man kann doch schon eine große Anzahl Ebenen auf denen etwas stattfindet, simulieren.

Damit in Basic alles schneller abläuft, sollten die Werte am Anfang des Programmes errechnet und in einem Array abgespeichert werden.

```
B (Ø) = 1
FOR I = 1 TO 19
B (I) = B (I-1) * 1,25
NEXT
```

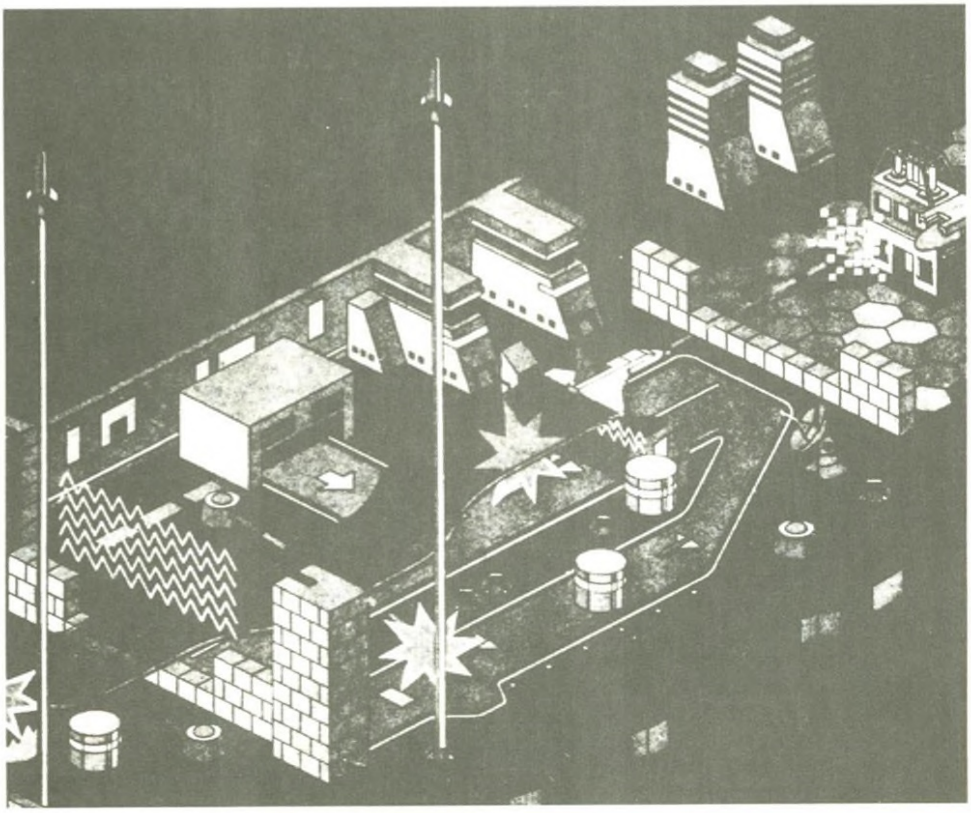
Um auch Darstellungen vor dem Bildschirm zu ermöglichen, muß jetzt noch folgende Modifikation programmiert werden.

```
FOR I = Ø TO 19
SB (I) = B(I)-B(1Ø)
NEXT
```

Im gleichen Faktor müssen auch die einzelnen Figuren erscheinen. Ein Raumschiff wurde nach obigem Beispiel in 20 facher Ausführung jeweils um 1,25 vergrößert in Sprites oder Shapes abgespeichert. Größe und Abstand simulieren bei Druck des Objektes in zwei Farben - wobei darauf zu achten ist, daß Farbüberschneidungen einen Purpurton ergeben müssen.

Doch nicht immer, wenn alles mathematisch richtig ist, stellt sich ein 3-D-Effekt ein. Das Gehirn muß laufend die Möglichkeit haben zu vergleichen. Es ist also wichtig, laufend verschiedene Ebenen besetzt zu haben, besonders den Hintergrund - sprich die entfernteste Ebene - vor der sich die Aktionen abspielen.

Im Moment arbeiten wir gerade an einigen echten 3-D-Programmen für verschiedene Rechner, die wir in folgenden CPU und Homecomputer vorstellen wollen.



Poker

für den TI-99/4A

Das Programm "Poker" hat nichts mit dem eigentlichen Kartenspiel Poker gemeinsam. Es wurde vielmehr einem Geldspielautomaten aus Las Vegas nachempfunden. Es ist für diejenigen, die versuchen ihr Geld zu vermehren, bzw. zu verlieren. Man beginnt mit einem Startkapital von 1 000 000 DM oder Dollar. Auf dem Bildschirm erscheinen dementsprechend Fragen oder Anweisungen, welche kurz erklärt werden:

Einsatz: Es kann ein Betrag zwischen 1 und 10 000 gesetzt werden. Es dürfen aber nur positive Beträge eingesetzt werden.
Start: Es kann jede Taste gedrückt werden.
Wieviel Karten halten?: Es können bis zu fünf Karten gehalten werden; eine muß jedoch mindestens aufgenommen werden.
Welche Karten?: Es muß darauf geachtet werden, daß zuerst die niedrigste Karte gehalten wird, z.B. die Karte "2" und "5" sollen gehalten werden. In diesem Fall muß zuerst eine 2 und dann die 5 gedrückt werden.
Neues Spiel: Hierfür kann jede Taste betätigt werden.
Um zum Ausgangsbild zurückzukehren muß man (FCTN=) drücken. Um zum Anfang des Spieles zu kommen, muß man (FCTN9) drücken.
Noch eines: Gewinnen kann man erst ab 2 Paaren.

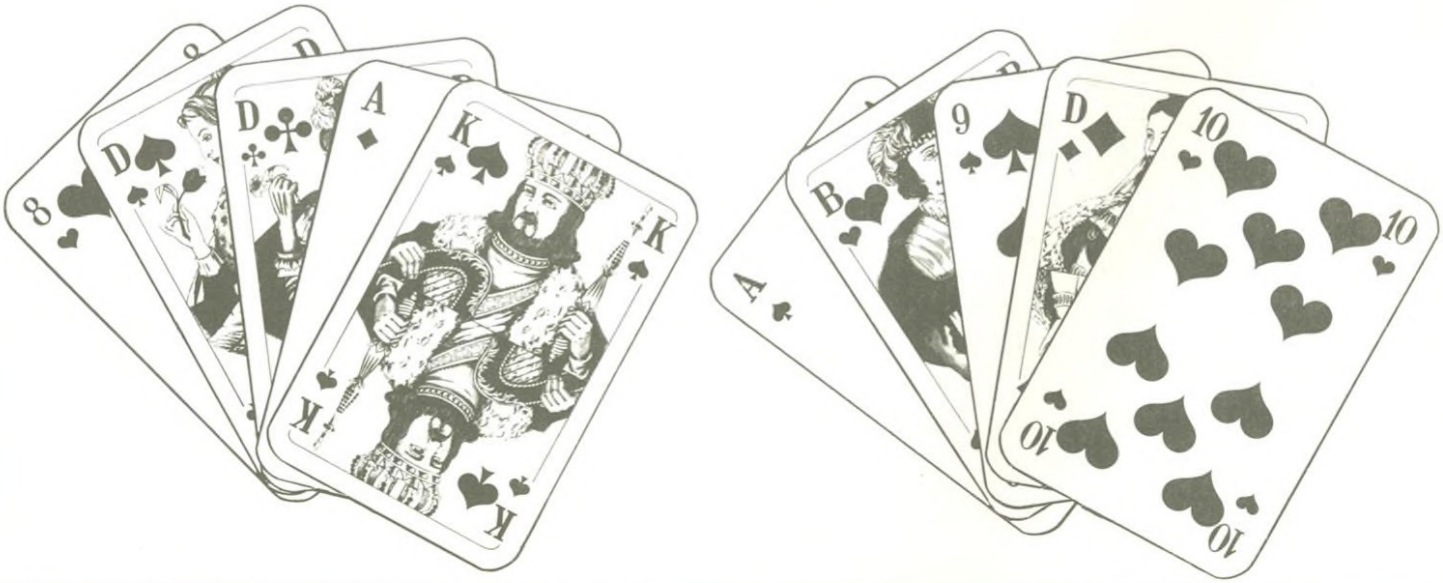
BLACKJACK

für TI 99/4A

Das Ziel dieses Kartenspieles ist es, mit höchstens 5 Karten möglichst nah an 21 Augen zu kommen. Hierbei tritt der Spieler gegen die Bank (Computer) an.

Spielbeschreibung:
Nachdem die Eingangsmelodie verklungen ist, beginnt der Computer die Graphik zu erstellen.
Wie man in der oberen linken Ecke des Bildschirmes ablesen kann, besitzt jeder Spieler ein Anfangskapital von 1000 Dollar. Der blinkende Cursor fordert den Wetteinsatz (nach der Eingabe "enter" drücken). Nachdem eine beliebige Summe eingegeben ist, deckt der Computer die ersten Karten auf. Hierbei erhält die Bank eine Karte verdeckt. Die Aufgabe des Spielers ist es nun, möglichst nah an 21 Punkte zu kommen. Zu diesem Zweck ist es möglich weitere Karten zu verlangen (W-Taste). Kommt er hierbei über 21 verliert er seinen Einsatz sofort.
Werden keine Karten mehr verlangt (K-Taste), deckt die Bank ihre zweite Spielkarte auf, und gibt sich gegebenenfalls weitere Karten. Dies tut sie solange, bis ihre Gesamtsumme "16" überschreitet. Auch bei einer "weichen 17" (unter den Karten befindet sich ein Ass, das 11 Punkte zählt) nimmt sie sich noch eine weitere Karte. Zuletzt ermittelt der Computer den Spieler, der der 21 am nächsten gekommen ist.
Noch eine weitere Besonderheit muß erwähnt werden: Gelingt es einem Spieler mit zwei Karten 21 Augen zu erhalten (z.B. Ass und 10), besitzt er einen "BLACKJACK", d.h. er erhält den 1,5 fachen Wetteinsatz. Ist ein Spieler so vom Pech verfolgt, daß er sein gesamtes Geld verspielt, erhält er wiederum ein Startkapital von 1000 Dollar.

Die Bewertung der Karten:
Die Karten von 1 - 9 zählen den aufgedruckten Wert.
Bildkarten und 10 zählen 10 Augen.
Sonderstellung des Asses: Ass zählt 1 oder 11 Punkte. Kommt ein Spieler über 21 Augen und er besitzt ein Ass, so zählt diese Karte nicht 11 sondern nur 1 Punkt.



```
10 REM ********************
20 REM *                  *
30 REM *     BLACKJACK    *
40 REM *                  *
50 REM *   RALF ROSSBACH  *
60 REM *   SCHULSTR. 46   *
70 REM *5205 ST. AUGUSTIN 1*
80 REM *                  *
90 REM ********************
100 CALL CLEAR
110 FOR L=9 TO 11 :: CALL COLOR(L,1,1):: NEXT L
120 CALL CHAR(96,"367F7F7F3E1C0800",104,"081C2A772A080800",112,"FFFFFFFFFFFFFFFF
",113,"0707070707070707",114,"E0E0E0E0E0E0E0E0")
130 CALL CHAR(105,"081C3E7F7F080800",64,"0044004444444438",35,"3C4299A1A199423C"
)
140 CALL SCREEN(11)
150 FOR I=1 TO 3 :: CALL VCHAR(3,4+I,112,7):: CALL VCHAR(3,24+I,112,7):: CALL VC
HAR(12,6+I,112,7):: CALL VCHAR(12,22+I,112,7):: NEXT I
160 CALL VCHAR(3,4,113,7):: CALL VCHAR(3,8,114,7):: CALL VCHAR(3,24,113,7):: CAL
L VCHAR(3,28,114,7)
170 CALL VCHAR(12,6,113,7):: CALL VCHAR(12,10,114,7):: CALL VCHAR(12,22,113,7):
 CALL VCHAR(12,26,114,7)
180 FOR I=1 TO 3 STEP 2 :: CALL HCHAR(4,4+I,96):: CALL HCHAR(8,4+I,96):: CALL HC
HAR(13+I,8,105):: NEXT I :: CALL HCHAR(6,6,96)
190 FOR I=1 TO 5 STEP 2 :: CALL VCHAR(3+I,26,104):: CALL VCHAR(12+I,23,96):: CAL
L VCHAR(12+I,25,96):: NEXT I
200 FOR I=1 TO 5 STEP 2 :: CALL VCHAR(12+I,7,105):: CALL VCHAR(12+I,9,105):: NEX
T I
210 CALL COLOR(9,9,16,10,2,16,11,16,1)
220 DISPLAY AT(4,10)SIZE(9):"BLACKJACK" :: DISPLAY AT(7,8)SIZE(13):"F@R  TI 99/4
A"
230 DISPLAY AT(22,2):"#1983 RALF ROSSBACH              SOFTWARE PRODUKTION"
240 DIM D(51),F(51)
250 I=I+1 :: IF I=52 THEN 360
260 READ D(I),F(I):: IF D(I)=0 THEN 350
270 IF D(I)=1 AND WE=1 THEN RESTORE 320 :: GOTO 250
280 IF D(I)=1 THEN RESTORE :: I=0 :: WE=1 :: GOTO 250
290 CALL SOUND(D(I),F(I),0):: GOTO 250
300 DATA 230,220,230,246,230,261,230,220,230,246,230,261,0,0,0,0,230,261,230,220
,230,246,230,261,0,0,0,0
310 DATA 230,246,230,220,230,261,280,261,500,220,0,0,0,0,0,0,1,1
320 DATA 280,220,500,349,230,293,500,329,0,0,0,0,0,0,230,349,230,329,230,349,230
,329,230,220,500,293
330 DATA 280,261,280,246,230,220,230,246,230,261,230,220,230,246,230,261,0,0,0,0
,230,246,230,220
340 DATA 230,261,280,261,600,220
350 FOR P=1 TO 25 :: NEXT P :: GOTO 250
360 CALL CLEAR :: CALL MAGNIFY(3)
370 CALL CHAR(58,"030F3FFFFFFFFFFF",59,"C0F0FCFFFFFFFFFF")
380 CALL CHAR(40,"00001F000000000000000000000000000000F800000000000000000")
390 CALL CHAR(60,"2010080700000000000000000000000040810E0000000000000000")
400 CALL CHAR(64,"FCFCFCFCFCFCFCFC",65,"3F3F3F3F3F3F3F3F",66,"FFFFFFFFFEFC0C0C",
67,"FCFCFC0000000078")
410 CALL CHAR(68,"3F3F3F000000001E",69,"FFFFFFFF7F3F3030",70,"0C0C0C0C00000000",
71,"303000040404040")
420 CALL CHAR(72,"0C0C002020202020",73,"3030303000000000",74,"0000000000000001
",75,"0407041C1C7070C0")
430 CALL CHAR(76,"20E02038380E0E03",77,"0000000000000080",78,"00",34,"0103070F1F
3F7FFF",35,"80C0E0F0F8FCFEFF")
440 CALL CHAR(80,"0000000000000303",81,"0303030303030303",82,"000000000000C0C0",
83,"C0C0C0C0C0C0C0C0")
450 CALL CHAR(84,"FF3F0F0F07010100",85,"FFFCF0F0E0808000")
460 CALL CHAR(134,"0080C0E3FFE3C080",135,"000103C7FFC70301",136,"C0C0C0C0C0C0C0C
0",137,"1010101010101010")
470 CALL CHAR(138,"0808080808080808",139,"181D1F1F1F1F1D18",140,"88D8F8F8F8F8D88
8",141,"88DDFFFFFFFFDD88")
480 CALL CHAR(142,"0000008080000000",143,"0303030303030303")
490 CALL CHAR(37,"FFFFFFFFFFFFFFFF",47,"FFFFFFFFFFFFFFFF",110,"8142241818244281"
)
500 FOR ZA=48 TO 57 :: CALL CHARPAT(ZA,ZA$):: CALL CHAR(ZA+40,ZA$):: CALL CHAR(Z
A+64,ZA$):: NEXT ZA
510 K$="0044485060504844" :: L$="0004040404044438" :: M$="0038444444544834" :: N
$="003844447C444444"
520 CALL CHAR(98,K$,99,L$,108,M$,109,N$)
530 CALL CHAR(100,"00000103070F1F3F",101,"1F0F070301000000",102,"0080C0E0F0F8FCF
E",103,"FCF8F0E0C0800000")
540 CALL CHAR(104,"00001C3E7F7F7F7F",105,"3F1F0F0703010000",106,"0000387CFEFEFEF
E",107,"FCF8F0E0C0800000")
550 CALL CHAR(122,K$,123,L$,132,M$,133,N$)
560 CALL CHAR(124,"0003030018181F18",125,"1800000000000300",126,"00E0E0808C8CFC8
C",127,"8C8080808080E000")
570 CALL CHAR(128,"00000103070F1F1F",129,"3F3F3E3D01010100",130,"00000080C0E0F0F
0",131,"F8F8787800000000")
580 CALL SCREEN(6):: FOR I=1 TO 14 :: CALL COLOR(I,1,1):: NEXT I
590 CALL HCHAR(1,15,58):: CALL HCHAR(1,18,59)
600 FOR I=1 TO 4 :: CALL HCHAR(2,14+I,65+I):: CALL HCHAR(3,14+I,69+I):: CALL HCH
AR(4,14+I,73+I):: NEXT I
610 FOR I=1 TO 2
620 CALL HCHAR(1+I,14,79+I):: CALL HCHAR(1+I,19,81+I):: CALL HCHAR(1,15+I,63+I)
: CALL HCHAR(6,15+I,133+I):: CALL HCHAR(5+I,14-I,34)
630 CALL HCHAR(5+I,19+I,35):: CALL HCHAR(10+I,15,37,4):: NEXT I
640 CALL HCHAR(5,16,78,2):: CALL HCHAR(5,15,84):: CALL HCHAR(5,18,85):: CALL HCH
AR(6,15,143):: CALL HCHAR(6,18,136):: CALL HCHAR(6,14,37)
650 CALL HCHAR(6,19,37):: CALL HCHAR(7,13,37,8):: CALL HCHAR(8,12,37,2):: CALL H
CHAR(9,12,141,2):: CALL HCHAR(8,14,138)
660 CALL HCHAR(8,19,137):: CALL HCHAR(8,20,37,2):: CALL HCHAR(9,14,140):: CALL H
CHAR(9,19,139):: CALL HCHAR(9,20,141,2)
670 FOR I=1 TO 3 :: CALL HCHAR(9+I,12,37,2):: CALL HCHAR(9+I,20,37,2):: CALL HCH
AR(7+I,15,37,2):: NEXT I
680 CALL VCHAR(10,14,138,3):: CALL VCHAR(10,19,137,3):: CALL VCHAR(8,17,142,3)::
 CALL VCHAR(8,18,37,3):: CALL SPRITE(#28,60,1,33,121)
690 CALL HCHAR(10,1,47,8):: CALL HCHAR(11,1,47,9):: CALL HCHAR(12,1,47,10)
700 CALL HCHAR(10,25,47,8):: CALL HCHAR(11,24,47,9):: CALL HCHAR(12,23,47,10)
710 CALL HCHAR(13,1,47,384):: CALL HCHAR(3,3,36):: CALL HCHAR(5,3,36)
720 CALL COLOR(1,16,1,2,13,1,3,2,1,4,2,1,5,2,10,6,2,10,7,10,1)
730 CALL COLOR(13,2,16,14,2,16,8,9,16,9,9,16,10,9,16,11,2,16,12,2,16):: CALL COL
OR(#28,7)
740 RANDOMIZE
750 DIM B(52):: SG=1000 :: GOSUB 1200
760 DISPLAY AT(3,2)SIZE(8):SG
770 WB,WS,A1,A2=0
780 IF N<=9 THEN GOSUB 1200
790 ACCEPT AT(5,3)SIZE(7)VALIDATE(DIGIT):ES :: IF ES>SG OR ES<=0 THEN 790
800 GOSUB 1240 :: IF NWK=11 THEN A1=A1+1
810 WB=NWK :: ZE=13 :: SP=24
820 CALL KARTEN(ZE,SP,WK,AK)
830 GOSUB 1240 :: WB=WB+NWK :: SP=28 :: IF NWK=11 THEN A1=A1+1
840 IF WB=21 THEN CALL KARTEN(ZE,SP,WK,AK):: BJ=1 :: GOTO 870 ELSE 850
850 FOR I=0 TO 2 :: CALL VCHAR(ZE,SP+I,110,5):: NEXT I :: M1K=WK :: M2K=AK
860 IF WB>21 THEN WB=WB-10 :: A1=1
870 FOR I=1 TO 2 :: GOSUB 1240 :: WS=WS+NWK :: IF NWK=11 THEN A2=A2+1
880 IF I=1 THEN SP=3 ELSE SP=7
890 CALL KARTEN(ZE,SP,WK,AK):: NEXT I
900 IF BJ=1 THEN 1160
910 IF WS=21 THEN 1190
920 IF WS>21 THEN WS=WS-10 :: A2=1
930 SP=3 :: ZE=19 :: Z=0
940 CALL KEY(0,C,S):: IF S=0 THEN 940
950 IF Z=3 AND C=87 THEN 940
960 IF C=87 THEN 980
970 IF C=75 THEN 1020 ELSE 940
980 GOSUB 1240 :: CALL KARTEN(ZE,SP,WK,AK):: WS=WS+NWK :: SP=SP+4 :: IF Z=1 THEN
 ZE=16
990 IF NWK=11 THEN A2=A2+1
1000 IF WS>21 AND A2>0 THEN WS=WS-10 :: A2=A2-1 :: GOTO 1000
1010 IF WS>21 THEN GOSUB 1100 :: CALL KARTEN(13,28,M1K,M2K):: GOTO 1120 ELSE Z=Z
+1 :: GOTO 940
1020 CALL KARTEN(13,28,M1K,M2K):: ZE=19 :: SP=32 :: Z=0
1030 IF Z<3 AND((WB<=17 AND A1>0)OR(WB<=16))THEN 1040 ELSE 1080
1040 GOSUB 1240 :: WB=WB+NWK :: SP=SP-4 :: IF NWK=11 THEN A1=A1+1
1050 CALL KARTEN(ZE,SP,WK,AK):: IF Z=1 THEN ZE=16
1060 IF WB>21 AND A1>0 THEN WB=WB-10 :: A1=A1-1 :: GOTO 1060
1070 IF WB>21 THEN GOSUB 1110 :: GOTO 1120 ELSE Z=Z+1 :: GOTO 1030
1080 IF WS>WB THEN GOSUB 1110 :: GOTO 1120
1090 IF WS<WB THEN GOSUB 1100 :: GOTO 1120 ELSE 1120
1100 SG=SG-ES :: DISPLAY AT(3,2)SIZE(8):SG :: CALL SOUND(500,246,0):: CALL SOUND
(200,220,0):: RETURN
1110 SG=SG+ES :: DISPLAY AT(3,2)SIZE(8):SG :: FOR T=320 TO 400 STEP 40 :: CALL S
OUND(400,T,0):: NEXT T :: RETURN
1120 DISPLAY AT(5,3)SIZE(8):" "
1130 FOR V=1 TO 1000 :: NEXT V
1140 CALL DELSPRITE(ALL):: CALL SPRITE(#27,40,7,33,121):: CALL HCHAR(13,1,47,384
)
1150 CALL PATTERN(#27,60):: IF SG=0 THEN 750 ELSE 770
1160 SG=SG-INT(1.5*ES):: IF SG<0 THEN SG=0
1170 FOR T=195 TO 135 STEP -20 :: CALL SOUND(350,T,0):: NEXT T :: CALL SOUND(450
,120,0)
1180 DISPLAY AT(3,2)SIZE(8):SG :: BJ=0 :: GOTO 1120
1190 RESTORE 300 :: FOR Z=1 TO 6 :: READ D(Z),F(Z):: CALL SOUND(D(Z),F(Z),0):: N
EXT Z :: SG=SG+INT(1.5*ES):: DISPLAY AT(3,2)SIZE(8):SG :: GOTO 1120
1200 T=0 :: N=52
1210 FOR I=1 TO 4
1220 FOR J=10 TO 130 STEP 10 :: T=T+1 :: B(T)=J+I :: NEXT J
1230 NEXT I :: RETURN
1240 R=INT(RND*N)+1 :: WK=INT(B(R)/10):: AK=B(R)-10*WK :: B(R)=B(N):: N=N-1
1250 IF WK=1 OR WK=12 OR WK=13 THEN NWK=10 ELSE NWK=WK
```

235

```
1260 RETURN
1270 SUB KARTEN(ROW,COL,WK,AK)
1280 WH=0 :: Y=ROW*8+15 :: X=COL*8 :: IF ANZAHL=0 OR ANZAHL=20 THEN ANZAHL=1 ELS
E ANZAHL=ANZAHL+1
1290 IF AK=1 OR AK=3 THEN GOSUB 1320 ELSE GOSUB 1390
1300 GOSUB 1460
1310 GOSUB 1500
1320 FARBE=9
1330 IF WK=1 THEN CODE1=108 :: RETURN
1340 IF WK=11 THEN CODE1=109 :: RETURN
1350 IF WK=12 THEN CODE1=98 :: RETURN
1360 IF WK=13 THEN CODE1=99 :: RETURN
1370 IF WK=10 THEN CODE1=89 :: WH=2 ELSE CODE1=WK+88
1380 RETURN
1390 FARBE=2
1400 IF WK=1 THEN CODE1=132 :: RETURN
1410 IF WK=11 THEN CODE1=133 :: RETURN
1420 IF WK=12 THEN CODE1=122 :: RETURN
1430 IF WK=13 THEN CODE1=123 :: RETURN
1440 IF WK=10 THEN CODE1=113 :: WH=2 ELSE CODE1=WK+112
1450 RETURN
1460 IF AK=1 THEN CODE2=104 :: RETURN
1470 IF AK=2 THEN CODE2=128 :: RETURN
1480 IF AK=3 THEN CODE2=100 ELSE CODE2=124
1490 RETURN
1500 CALL HCHAR(ROW,COL,CODE1):: IF WH=2 THEN CALL HCHAR(ROW,COL+1,CODE1-1):: CA
LL HCHAR(ROW,COL+2,37)ELSE CALL HCHAR(ROW,COL+1,37,2)
1510 FOR I=0 TO 2 :: CALL VCHAR(ROW+1,COL+I,37,4):: NEXT I :: CALL SPRITE(#ANZAH
L,CODE2,FARBE,Y,X):: SUBEND
```

235

```
1 REM ****************************************************
2 REM                          P O K E R
3 REM ****************************************************
4 REM
5 CALL CLEAR :: CALL SCREEN(11)
6 FOR A=1 TO 14 :: CALL COLOR(A,2,1):: NEXT A
7 CALL CHARSET
10 CALL CHAR(97,"00441028447C4444",111,"0044384444444438",117,"0044004444444438"
)
15 DISPLAY AT(10,1):"SPIELANLEITUNG ERWuNSCHT ?" :: DISPLAY AT(12,1):"          (J
/N)"
20 CALL KEY(3,K,S):: IF K<>74 AND K<>78 THEN 20 ELSE IF K=78 THEN CALL CLEAR ::
GOTO 80
25 DISPLAY AT(7,1):"WENN DAS WORT EINSATZ AUF   DEM BILDSCHIRM ERSCHEINT,   KANN
 EIN BETRAG ZWISCHEN 1  UND 10000 GESETZT WERDEN."
30 DISPLAY AT(11,1):"DABEI DuRFEN NUR POSITIVE   BETRaGE GESETZT WERDEN." :: GOS
UB 1285
35 DISPLAY AT(7,1):"UM DAS SPIEL ZU STARTEN     KANN JEDE BELIEBIGE TASTE   GEDR
uCKT WERDEN." :: GOSUB 1285
40 DISPLAY AT(7,1):"BEI DER FRAGE               'WIEVIELE KARTEN HALTEN ?'  KoNN
EN BIS FuNF KARTEN GE-  HALTEN WERDEN; ES MUSS JE-"
45 DISPLAY AT(11,1):"DOCH MINDESTENS EINE GEHAL- TEN WERDEN." :: GOSUB 1285
50 DISPLAY AT(7,1):"BEIM KARTENHALTEN MUSS      DARAUF GEACHTET WERDEN,     DASS
 ZUERST DIE NIEDRIG-    STE KARTE GEHALTEN WIRD."
55 DISPLAY AT(11,1):"BEISPIEL:"
60 DISPLAY AT(12,1):"DIE KARTEN '2' UND '5'      SOLLEN GEHALTEN WERDEN.     IN
DIESEM FALL WIRD ZU-     ERST DIE '2' GEDRuCKT       UND DANN DIE '5'." :: GOSUB
 1285
65 DISPLAY AT(7,1):"GEWINNEN KANN MAN ERST,     WENN MAN MINDESTENS ZWEI    PAAR
E HAT."
70 DISPLAY AT(11,1):"AUF DIE FRAGE 'NEUES SPIEL' KANN JEDE TASTE GEDRuCKT    WER
DEN. UM ZUM AUSGANGS-    BILD ZU GELANGEN MUSS       (FCTN=) GEDRuCKT WERDEN."
75 DISPLAY AT(16,1):"UM ZUM ANFANG ZU GELANGEN   MUSS (FCTN9) GEDRuCKT WER-  DEN
." :: GOSUB 1285
80 CALL SCREEN(2)
85 OPTION BASE 1
90 DIM Z(4,13)
95 G$="00003F3F303030303F3F303030303030000C0C030303030C0C"
100 G1$="00003F3F303030303030303030303F3F0000F0F0303030303030303030303030F0F0"
105 G2$="0000303030303333C3C3333303030300000303 0C0C0000000000000C0C03030"
110 G3$="00003F3F303030303F3F303030303F3F0000F0F000000000C0C000000000F0F0"
115 G4$="00003F3F303030303F3F3333303030300000C0C030303030C0C00000C0C03030"
120 CALL CHAR(104,G$):: CALL CHAR(108,G1$):: CALL CHAR(40,G2$):: CALL CHAR(44,G3
$):: CALL CHAR(100,G4$)
125 CALL SPRITE(#1,104,16,72,24):: CALL SPRITE(#2,108,11,104,72):: CALL SPRITE(#
3,40,7,72,120)
130 CALL SPRITE(#4,44,13,104,168):: CALL SPRITE(#5,100,5,72,216):: CALL MAGNIFY(
4)
135 FOR A=1 TO 100 :: NEXT A
140 FOR A=1 TO 5 :: RANDOMIZE :: CALL MOTION(#A,10+A,0):: NEXT A
145 CALL CHAR(60,"00FF")
150 CALL CHAR(112,"9942249999244299")
155 CALL CHAR(120,"FFFFFFFFFFFFFFFF")
160 CALL CHAR(128,"66FFFFFFFF7E3C18")
165 CALL CHAR(129,"183C7EFFFF7E3C18")
170 CALL CHAR(136,"383810D6FED61038")
175 CALL CHAR(137,"1054FEFEFE7C381")
180 FOR A=3 TO 8 :: CALL COLOR(A,2,2):: NEXT A
185 CALL COLOR(11,2,2):: CALL COLOR(12,16,1):: CALL COLOR(13,7,16):: CALL COLOR(
14,2,16)
190 FOR Q=1 TO 4
195 FOR C=1 TO 13 :: P=P+1
200 Z(Q,C)=P :: NEXT C
205 NEXT Q
210 H1=1000000
215 DISPLAY AT(3,4):"EINSATZ:"
220 DISPLAY AT(3,20):"GUTHABEN:"
225 DISPLAY AT(5,21):H1
230 E1=0 :: X=0 :: P3=0
235 RESTORE 240 :: FOR A=1 TO 20 :: READ V
240 DATA 3,4,5,6,9,10,11,12,15,16,17,18,21,22,23,24,27,28,29,30
245 CALL VCHAR(9,V,112,7):: NEXT A
250 CALL DELSPRITE(ALL):: CALL COLOR(11,16,5)
255 FOR A=3 TO 8 :: CALL COLOR(A,15,1):: NEXT A
260 DISPLAY AT(21,2):"EINSATZ ?" :: ACCEPT AT(21,13)SIZE(6)VALIDATE(DIGIT):E1
265 ON ERROR 275
270 IF E1>10000 XOR E1<1 THEN 275 ELSE 285
275 DISPLAY AT(21,2):"ES KOENNEN NUR GANZE" :: DISPLAY AT(22,2):"ZAHLEN BIS ZU 1
0000" :: DISPLAY AT(23,2):"DM GESETZT WERDEN."
280 FOR A=1 TO 500 :: NEXT A :: CALL HCHAR(21,1,32,96):: GOTO 260
285 DISPLAY AT(5,6):E1
290 H1=H1-E1 :: DISPLAY AT(5,21):H1
295 DISPLAY AT(23,6):"          "
300 DISPLAY AT(21,2):"SPIEL STARTEN !"
305 CALL KEY(0,K,S)
310 IF S=0 THEN 305
315 DISPLAY AT(21,2):"                    "
320 RANDOMIZE
325 FOR A=1 TO 5
330 F(A)=INT((RND*10)+1)
335 IF F(A)>4 THEN 330
340 B(A)=INT((RND*20)+1)
345 IF B(A)>13 THEN 340
350 ON A GOTO 355,365,375,385,395
355 D1=Z(F(A),B(A))
360 GOTO 405
365 D2=Z(F(A),B(A))
370 IF D2=D1 THEN 330 ELSE 405
375 D3=Z(F(A),B(A))
380 IF D3=D1 XOR D3=D2 THEN 330 ELSE 405
385 D4=Z(F(A),B(A))
390 IF D4=D1 XOR D4=D2 XOR D4=D3 THEN 330 ELSE 405
395 D5=Z(F(A),B(A))
400 IF D5=D1 XOR D5=D2 XOR D5=D3 XOR D5=D4 THEN 330
405 NEXT A
410 H=10
415 GOSUB 445
420 GOSUB 465
425 GOSUB 485
430 GOSUB 505
435 GOSUB 525
440 GOTO 565
445 V=3 :: O=B(1)
450 FOR A=1 TO 4 :: CALL VCHAR(9,2+A,120,7):: NEXT A
455 IF F(1)=1 THEN W=128 ELSE IF F(1)=2 THEN W=129 ELSE IF F(1)=3 THEN W=136 ELS
E W=137
460 GOTO 540
465 V=9 :: O=B(2)
470 FOR A=1 TO 4 :: CALL VCHAR(9,8+A,120,7):: NEXT A
475 IF F(2)=1 THEN W=128 ELSE IF F(2)=2 THEN W=129 ELSE IF F(2)=3 THEN W=136 ELS
E W=137
480 GOTO 540
485 V=15 :: O=B(3)
490 FOR A=1 TO 4 :: CALL VCHAR(9,14+A,120,7):: NEXT A
495 IF F(3)=1 THEN W=128 ELSE IF F(3)=2 THEN W=129 ELSE IF F(3)=3 THEN W=136 ELS
E W=137
500 GOTO 540
505 V=21 :: O=B(4)
510 FOR A=1 TO 4 :: CALL VCHAR(9,20+A,120,7):: NEXT A
515 IF F(4)=1 THEN W=128 ELSE IF F(4)=2 THEN W=129 ELSE IF F(4)=3 THEN W=136 ELS
E W=137
520 GOTO 540
525 V=27 :: O=B(5)
```

```
530 FOR A=1 TO 4 :: CALL VCHAR(9,26+A,120,7):: NEXT A
535 IF F(5)=1 THEN W=128 ELSE IF F(5)=2 THEN W=129 ELSE IF F(5)=3 THEN W=136 ELS
E W=137
540 IF O=1 XOR O>9 THEN 550
545 CALL HCHAR(H,V+1,W,1):: DISPLAY AT(H+3,V-1)SIZE(1):USING "#":O :: RETURN
550 IF O=11 THEN J$="B" ELSE IF O=12 THEN J$="D" ELSE IF O=13 THEN J$="K" ELSE I
F O=1 THEN J$="A" ELSE 560
555 CALL HCHAR(H,V+1,W,1):: DISPLAY AT(H+3,V-1)SIZE(1):USING "#":J$ :: RETURN
560 CALL HCHAR(H,V+1,W,1):: DISPLAY AT(H+3,V-1)SIZE(2):USING "##":O :: RETURN
565 DISPLAY AT(21,2):"WIEVIELE KARTEN" :: DISPLAY AT(23,9):"HALTEN ?"
570 CALL KEY(0,K,S)
575 IF S=0 THEN 570 ELSE IF K<49 OR K>53 THEN 570
580 DISPLAY AT(23,9):"          "
585 DISPLAY AT(21,2):"WELCHE KARTEN ?"
590 FOR A=49 TO K
595 CALL KEY(0,K,S)
600 IF S=0 THEN 595 ELSE IF K=49 THEN 615 ELSE IF K=50 THEN 620 ELSE IF K=51 THE
N 625 ELSE IF K=52 THEN 630 ELSE IF K=53 THEN 635 ELSE 595
605 NEXT A
610 DISPLAY AT(21,2):"                    " :: GOTO 640
615 DISPLAY AT(17,1):"HOLD" :: CALL HCHAR(18,3,60,4):: GOTO 605
620 DISPLAY AT(17,7):"HOLD" :: CALL HCHAR(18,9,60,4):: GOTO 605
625 DISPLAY AT(17,13):"HOLD" :: CALL HCHAR(18,15,60,4):: GOTO 605
630 DISPLAY AT(17,19):"HOLD" :: CALL HCHAR(18,21,60,4):: GOTO 605
635 DISPLAY AT(17,25):"HOLD" :: CALL HCHAR(18,27,60,4):: GOTO 605
640 FOR V=3 TO 27 STEP 6
645 CALL GCHAR(17,V,P1)
650 IF P1=32 THEN 665
655 NEXT V
660 GOTO 675
665 FOR A=1 TO 4 :: CALL VCHAR(9,(V-1)+A,112,7):: NEXT A
670 GOTO 655
675 FOR V=3 TO 27 STEP 6
680 CALL GCHAR(9,V,P2)
685 IF P2=112 THEN 700
690 NEXT V
695 GOTO 835
700 RANDOMIZE :: P3=P3+1
705 O=INT((RND*10)+1)
710 IF O>4 THEN 705
715 O=INT((RND*20)+1)
720 IF O>13 THEN 715
725 IF P3=1 THEN 730 ELSE IF P3=2 THEN 755 ELSE IF P3=3 THEN 780 ELSE IF P3=4 TH
EN 805
730 D6=Z(O,O)
735 IF D6=D1 XOR D6=D2 XOR D6=D3 XOR D6=4 XOR D6=D5 THEN 705 ELSE GOSUB 830
740 IF O=1 THEN W=128 ELSE IF O=2 THEN W=129 ELSE IF O=3 THEN W=136 ELSE W=137
745 GOSUB 540
750 GOTO 690
755 D7=Z(O,O)
760 IF D7=D1 XOR D7=D2 XOR D7=D3 XOR D7=D4 XOR D7=D5 XOR D7=D6 THEN 705 ELSE GOS
UB 830
765 IF O=1 THEN W=128 ELSE IF O=2 THEN W=129 ELSE IF O=3 THEN W=136 ELSE W=137
770 GOSUB 540
775 GOTO 690
780 D8=Z(O,O)
785 IF D8=D1 XOR D8=D2 XOR D8=D3 XOR D8=D4 XOR D8=D5 XOR D8=D6 XOR D8=D7 THEN 70
5 ELSE GOSUB 830
790 IF O=1 THEN W=128 ELSE IF O=2 THEN W=129 ELSE IF O=3 THEN W=136 ELSE W=137
795 GOSUB 540
800 GOTO 690
805 D9=Z(O,O)
810 IF D9=D1 XOR D9=D2 XOR D9=D3 XOR D9=D4 XOR D9=D5 XOR D9=D6 XOR D9=D7 XOR D9=
D8 THEN 705 ELSE GOSUB 830
815 IF O=1 THEN W=128 ELSE IF O=2 THEN W=129 ELSE IF O=3 THEN W=136 ELSE W=137
820 GOSUB 540
825 GOTO 690
830 FOR A=1 TO 4 :: CALL VCHAR(9,(V-1)+A,120,7):: NEXT A :: RETURN
835 FOR A=1 TO 5
840 IF A=1 THEN T=4 ELSE IF A=2 THEN T=10 ELSE IF A=3 THEN T=16 ELSE IF A=4 THEN
 T=22 ELSE IF A=5 THEN T=28
845 CALL GCHAR(13,T,P):: IF P=65 THEN P=49 ELSE IF P=49 THEN P=58 ELSE IF P=66 T
HEN P=59 ELSE IF P=68 THEN P=60 ELSE IF P=75 THEN P=61
850 B(A)=P :: NEXT A
855 FOR A=1 TO 5
860 IF A=1 THEN T=4 ELSE IF A=2 THEN T=10 ELSE IF A=3 THEN T=16 ELSE IF A=4 THEN
 T=22 ELSE IF A=5 THEN T=28
865 CALL GCHAR(10,T,P):: F(A)=P :: NEXT A
870 FOR A=1 TO 4
875 IF B(A)=B(A+1)THEN X=X+1
880 NEXT A
885 FOR A=1 TO 3
890 IF B(A)=B(A+2)THEN X=X+1
895 NEXT A
900 FOR A=1 TO 2
905 IF B(A)=B(A+3)THEN X=X+1
910 NEXT A
915 IF B(1)=B(5)THEN X=X+1
920 IF X=2 THEN 1120 ELSE IF X=3 THEN 1140 ELSE IF X=4 THEN 1180 ELSE IF X=6 THE
N 1160 ELSE IF X=1 THEN 1220
925 FOR A=1 TO 5
930 X=1
935 FOR C=1 TO 5
940 IF A=C THEN 950
945 IF B(A)>B(C)THEN X=X+1
950 NEXT C
955 L(A)=X
960 NEXT A
965 FOR A=1 TO 5
970 FOR C=1 TO 5
975 IF L(A)=C THEN B(11-C)=B(A)
980 NEXT C
985 NEXT A
990 X=0
995 M1=B(6) :: M2=B(7) :: M3=B(8) :: M4=B(9) :: M5=B(10)
1000 IF M1-1=M2 AND M2-1=M3 AND M3-1=M4 AND M4-1=M5 THEN X=4
1005 Y=0
1010 FOR A=1 TO 4
1015 IF F(A)=F(A+1)THEN Y=Y+1
1020 NEXT A
1025 IF B(6)=61 AND B(7)=60 AND B(8)=59 AND B(9)=58 AND B(10)=49 THEN 1035
1030 IF X=4 AND Y=4 THEN 1080 ELSE IF X=4 AND Y<4 THEN 1200 ELSE IF X<4 AND Y=4
THEN 1100 ELSE 1220
1035 IF Y=4 THEN 1060
1040 DISPLAY AT(21,2):"GROSSE STRASSE"
1045 E1=E1*5
1050 FOR A=1 TO 500 :: NEXT A
1055 GOTO 1235
1060 DISPLAY AT(21,2):"ROY FLASH"
1065 E1=E1*15
1070 FOR A=1 TO 500 :: NEXT A
1075 GOTO 1235
1080 DISPLAY AT(21,2):"STREET FLASH"
1085 E1=E1*8
1090 FOR A=1 TO 500 :: NEXT A
1095 GOTO 1235
1100 DISPLAY AT(21,2):"FLASH"
1105 E1=E1*6
1110 FOR A=1 TO 500 :: NEXT A
1115 GOTO 1235
1120 DISPLAY AT(21,2):"ZWEI PAARE"
1125 E1=E1*2
1130 FOR A=1 TO 500 :: NEXT A
1135 GOTO 1235
1140 DISPLAY AT(21,2):"DRILLING"
1145 E1=E1*4
1150 FOR A=1 TO 500 :: NEXT A
1155 GOTO 1235
1160 DISPLAY AT(21,2):"VIERLING"
1165 E1=E1*10
1170 FOR A=1 TO 500 :: NEXT A
1175 GOTO 1235
1180 DISPLAY AT(21,2):"FULL HOUSE"
1185 E1=E1*7
1190 FOR A=1 TO 500 :: NEXT A
1195 GOTO 1235
1200 DISPLAY AT(21,2):"KLEINE STRASSE"
1205 E1=E1*3
1210 FOR A=1 TO 500 :: NEXT A
1215 GOTO 1235
1220 DISPLAY AT(21,2):"VERLOREN"
1225 FOR A=1 TO 500 :: NEXT A
1230 GOTO 1255
1235 DISPLAY AT(21,2):"GEWINN:"
1240 DISPLAY AT(21,10):E1
1245 FOR A=1 TO 500 :: NEXT A
1250 H1=H1+E1 :: DISPLAY AT(5,21):H1
1255 DISPLAY AT(21,2):"NEUES SPIEL ?"
1260 CALL HCHAR(17,1,32,64)
1265 CALL KEY(3,K,S)
1270 IF S=0 THEN 1265 ELSE IF K=15 THEN 5
1275 DISPLAY AT(21,2):"                    " :: CALL HCHAR(9,1,32,224):: CALL COLOR(11,
2,2)
1280 DISPLAY AT(5,4)SIZE(8):"      " :: GOTO 230
1285 DISPLAY AT(24,1):"ENTER DRuCKEN" :: CALL KEY(3,K,S):: IF K<>13 THEN 1285 EL
SE CALL CLEAR :: RETURN
```

Vokabeltraining

Für den TI-99 4A + Ext. Basic

Wer wünscht sich nicht, perfect eine fremde Sprache zu beherrschen oder zumindest, sich einigermaßen in unseren Nachbarländern verständigen zu können. Ja, da könnte man wirklich stolz drauf sein. Aber jeder der diesen Versuch schon einmal unternommen hat oder gerade dabei ist eine andere Sprache zu erlernen, sei es Spanisch, Italienisch, Englisch, Französisch u.s.w., hat wohl sehr schnell gemerkt - ohne das konsequente Erlernen von Vokabeln geht es einfach nicht.

Dies kann man erstens allein versuchen, macht aber meistens keinen allzu großen Spaß. Zweitens: Freunde oder Verwandte mit der ehrenvollen Aufgabe betrauen, Ihnen die Vokabeln abhören zu dürfen, die dann natürlich nur darauf warten, Sie verbessern zu können (ist ja nur menschlich) und 3. Ihren Computer zur Hilfe nehmen - für viele wahrscheinlich die interessanteste und ausgefeilteste Lösung.
Auf jeden Fall wünschen wir Ihnen mit dem Programm Vokabeltraining Spaß und viel Erfolg!

Zum Programm:
Nach Erscheinen des Menues wählt man Taste 1, 2, 3, 4, oder 5.
Wollen Sie nun Vokabeln eingeben, drücken Sie Taste 1. Der Computer erwartet dann Ihre Eingabe von Vokabeln und Übersetzungen, jeweils 30. Wenn er fertig ist, zeigt er wieder das Menue.
Wollen Sie Vokabeln abfragen, drükken Sie 2. Geben Sie ein, wie oft Sie abgefragt werden möchten. Der Computer ermittelt per Zufall eine Vokabel, die er auf dem Bildschirm anzeigt. Geben Sie jetzt die Übersetzung ein! Ist die Übersetzung richtig, wird dies angezeigt und fortgefahren. Ist sie falsch, wird dies ebenfalls angezeigt und fortgefahren. Ist die gewünschte Anzahl Abfragen erreicht, errechnet der Computer, ob Sie mindenstens 5/6 der Abfragen richtig beantwortet haben; wenn ja, wird dies angezeigt und zum Menue zurückgegangen, wenn nein, müssen Sie mit anderen Vokabeln wiederholen.
Wenn Sie einlesen oder speichern wollen, drücken Sie 3 oder 4.
Der Computer gibt Ihnen dann weitere Instruktionen.

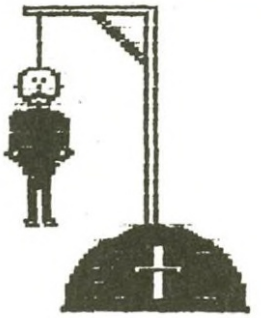
```
100 REM VOKABELTRAINING
110 CALL CLEAR
120 DIM V$(31,2)
130 DISPLAY AT(3,5):"VOKABEL
TRAINING"
140 DISPLAY AT(6,5):"BITTE W
AEHLEN SIE:" ::DISPLAY AT(8
,5):"1 FUER VOK. EINGEBEN" :
: DISPLAY AT(10,5):"2 FUER V
OK. ABFRAGEN"
150 DISPLAY AT(12,5):"3 FUER
 VOK. EINLESEN": :"    4 FUE
R VOK. SPEICHERN": :"    5 F
UER BEENDEN": : : : :"    IH
RE WAHL? "
160 ACCEPT AT(21,15)SIZE(-1)
BEEP VALIDATE("12345"):WAHL$
170 CALL CLEAR
180 IF WAHL$=" " THEN 130
190 IF VAL(WAHL$)=5 THEN END
200 ON VAL(WAHL$)GOSUB 220,3
10,500,580
210 GOTO 110
220 REM VOK. EINGEBEN
230 DISPLAY AT(3,5):"VOKABEL
N EINGEBEN": : :"    VOKABEL
" :: DISPLAY AT(14,5):"UEBER
SETZUNG"
240 FOR Z=1 TO 30
250 DISPLAY AT(6,13):Z
260 ACCEPT AT(8,5)BEEP:V$(Z,
1)
270 DISPLAY AT(14,18):Z
280 ACCEPT AT(16,5)BEEP:V$(Z
,2)
290 NEXT Z
300 RETURN
310 REM ABFRAGEN
320 DISPLAY AT(3,5):"VOKABEL
N ABFRAGEN": : :"    VOKABEL
" :: DISPLAY AT(14,5):"UEBER
SETZUNG"
330 RANDOMIZE
340 DISPLAY AT(24,5):"ANZAHL
 DER ABFRAGEN" :: ACCEPT AT(
24,25)BEEP VALIDATE(DIGIT):A
NZ
350 DISPLAY AT(24,5):"" :: R
=0
360 FOR Z=1 TO ANZ
370 X=INT(30*RND)+1
380 DISPLAY AT(8,5):V$(X,1)
390 ACCEPT AT(16,5)BEEP:EING
$
400 IF EING$=V$(X,2)THEN 410
 ELSE 420
410 DISPLAY AT(24,5):"RICHTI
G" :: FOR ZS=1 TO 999 :: NEX
T ZS :: R=R+1 :: GOTO 440
420 DISPLAY AT(24,5):"FALSCH
"
430 FOR ZS=1 TO 999 :: NEXT
ZS
440 DISPLAY AT(24,5):""
450 NEXT Z
460 IF R<5/6*ANZ THEN 470 EL
SE 480
470 DISPLAY AT(24,5):"BITTE
WIEDERHOLEN" :: FOR ZS=1 TO
999 :: NEXT ZS :: GOTO 350
480 DISPLAY AT(23,5):"SEHR G
UT:";R;"VON";ANZ;"    RICHTI
G" :: FOR ZS=1 TO 999 :: NEX
T ZS
490 RETURN
500 REM EINLESEN
510 OPEN #1:"CS1",INPUT ,INT
ERNAL,FIXED(64)
520 FOR Y=1 TO 30
530 INPUT #1:V$(Y,1)
540 INPUT #1:V$(Y,2)
550 NEXT Y
560 CLOSE #1
570 RETURN
580 REM SPEICHERN
590 OPEN #1:"CS1",OUTPUT,INT
ERNAL,FIXED(64)
600 FOR Y=1 TO 30
610 PRINT #1:V$(Y,1)
620 PRINT #1:V$(y,2)
630 NEXT Y
640 CLOSE #1
650 RETURN
```

****

Anmerkung:
ersetzt das Nummernzeichen

Hangman

für den TI-99/4A + Ext. Basic



Das Spiel Hangman ist die Computerversion des bekannten Ratespieles Galgenmännchen. Es kann mit zwei Personen gespielt werden, wobei einer ein Wort in den Computer eingibt und der andere dieses Wort erraten muß.

```
100 REM  HANGMAN
110 REM
120 REM  01.11.1983 BY
130 REM
140 REM  HEICO HOMBURG
150 REM
160 CALL CHAR(129,"FFFFFFFFF
FFFFFFF")
170 CALL CHAR(130,"OFOFOFOF"
)
180 CALL CHAR(131,"OFOFOFOFF
OFOFOFO")
190 CALL CHAR(132,"FFFFFFFF"
)
200 CALL CHAR(133,"OOOOOOOOF
OFOFOFO")
210 CALL CHAR(134,"FOFOFOFOF
OFOFOFO")
220 CALL CHAR(135,"FFFFFFFFF
OFOFOFO")
230 CALL CHAR(136,"FFFFFFFFO
FOFOFOF")
240 CALL CHAR(137,"FOFOFOFOO
FOFOFOF")
250 CALL CLEAR
260 X$=""
270 DISPLAY AT(2,11):"HANGMA
N" :: DISPLAY AT(4,1):"WORT?
" :: ACCEPT AT(6,1)SIZE(28)B
EEP:W$
280 CALL CLEAR
290 A=9
300 DISPLAY AT(2,1):"NOCH ";
STR$(A);" VERSUCHE"
310 S=11
320 DISPLAY AT(4,1):"GEBRAUC
HT:"
330 DISPLAY AT(6,1):"BUCHSTA
BE?"
340 FOR I=1 TO LEN(W$)
350 CALL HCHAR(8,I+2,45)
360 NEXT I
370 ACCEPT AT(6,12)SIZE(1)BE
EP:B$
380 DISPLAY AT(4,S):B$
390 S=S+1
400 B=0
410 FOR I=1 TO LEN(W$)
420 IF SEG$(W$,I,1)=B$ THEN
GOSUB 530
430 NEXT I
440 IF B=1 THEN GOSUB 880
450 IF B=1 THEN 370
460 A=A-1
470 DISPLAY AT(2,6)SIZE(1):S
TR$(A)
480 IF A=0 THEN 780
490 ON A GOSUB 750,720,700,6
80,660,620,590,560
500 DISPLAY AT(2,6)SIZE(1):S
TR$(A)
510 S=S+1
520 GOTO 370
530 CALL HCHAR(8,I+2,ASC(B$)
)
540 B=1
550 RETURN
560 DISPLAY AT(21,1):"B"
570 DISPLAY AT(20,2):"C"
580 RETURN
590 DISPLAY AT(21,4):"B"
600 DISPLAY AT(20,3):"DE"
610 RETURN
620 FOR X=15 TO 19
630 DISPLAY AT(X,3):"F"
640 NEXT X
650 RETURN
660 DISPLAY AT(15,3):"GDDD"
670 RETURN
680 DISPLAY AT(15,6):"H"
690 RETURN
700 DISPLAY AT(16,6):"O"
710 RETURN
720 DISPLAY AT(17,6):"A"
730 DISPLAY AT(18,6):"A"
740 RETURN
750 DISPLAY AT(17,5)SIZE(1):
"C"
760 DISPLAY AT(17,7):"I"
770 RETURN
780 DISPLAY AT(19,5):"C I"
790 DISPLAY AT(17,17):"AAAAA
AAAAA" :: DISPLAY AT(18,17):
"AERHANGENA" :: DISPLAY AT(1
9,17):"AAAAAAAAAA"
800 CALL SOUND(1000,131,0)
810 DISPLAY AT(10,1):"ES WAR
:" :: DISPLAY AT(12,1):W$
820 DISPLAY AT(23,1):"EIN NE
UES SPIEL? (J/N)"
830 CALL KEY(0,K,ST)
840 IF ST=0 THEN 830
850 IF K=74 THEN 250
860 IF K<>78 THEN 830
870 END
880 X$=""
890 FOR Y=1 TO LEN(W$)
900 CALL GCHAR(8,Y+2,Z)
910 X$=X$&CHR$(Z)
920 NEXT Y
930 IF W$=X$ THEN 950
940 RETURN
950 DISPLAY AT(17,17):"AAAAA
AA" :: DISPLAY AT(18,17):"AB
RAVOA" :: DISPLAY AT(19,17):
"AAAAAAA"
960 CALL SOUND(1000,440,0)
970 GOTO 820
```

Herzlichen Glückwunsch zu Ihrer wirklich guten Zeitschrift. Ich gehöre von Anfang an zu Ihrem Leserkreis. Früher besaß ich nur den ZX-81, jetzt habe ich mir auch den SPECTRUM zugelegt. Logisch, daß mich am meisten Ihre Spektrum-Programme interessieren. Meine Englischkenntnisse sind begrenzt und mit einem mir vorliegenden Handbuch, ist nicht viel Staat zu machen. So ist es mir bis heute noch nicht gelungen, richtig in die Geheimnisse der Sonderzeichen einzudringen. Sie druckten im Heft 6 das Frogger-Programm ab. Ich gab es ein, aber die Sonderzeichen wurden nicht gelesen. Trotz mehrmaligem Durchschauen, konnte ich keinen Eintippfehler entdecken. Ich wartete auf Heft 7, und damit auf eine eventuelle Berichtigung. Es kam aber keine, stattdessen ein neues Programm - Mampfman - und wieder taucht bei mir dasselbe Problem auf. Ich weiß nun nicht, ob der Fehler bei mir, oder im Programm liegt. Bei Durchsicht von entsprechender Literatur, wird jeweils anders vorgegangen. Ich wäre Ihnen dankbar, wenn Sie mir mitteilen würden, ob der Fehler bei mir, oder im Programm liegt.

F.-J. Grünen, Mainz

Redaktion: Bezüglich Ihrer Frage zum Frogger-Programm aus Heft 6: In den Zeilen 9300 bis 9600, werden die entsprechenden Grafikzeichen, Frosch, Schildkröten, Bäume, Autos, erzeugt. Auf die gleiche Art und Weise, werden Umlaute, zusätzlich zum normalen Zeichensatz, eingefügt. Durch Vergleich der Datas und der in den anhängenden REM's befindlichen Zeichen, können Sie durch einfache Schlußfolgerung, hinter das Geheimnis der Zeichensatzprogrammmierung, des ZX Spectrum, kommen. Im weiteren Programm auftauchende grafische Zeichen, müssen mit den entsprechenden Buchstaben, von der Tastatur, eingegeben werden.

Ich habe folgende Frage an Sie: Wie kann man Programme, die auf 8 K geschrieben wurden, auf 16 K umschreiben, ohne die Pokes zu ändern? Ausserdem möchte ich Sie fragen , wie man selber Zeichen für den VC–20 definieren kann (bin Anfänger!)

P. Rebmann, Pratteln (CH)

Redaktion: Zu Ihrer ersten Frage: Bei diesen Programmen muß man nichts umschreiben, es sei denn, daß Maschinenprogramme verwendet werden, die in einem anderen Speicherbereich nicht funktionstüchtig sind. Zweitens: In den Heften Homecomputer 3 und 4 haben wir die Programme Disassembler und Assembler abgedruckt, die sich ausführlich mit Ihrer Frage beschäftigen.

Zum Programm, Vokabeltraining für TI 99/4A, habe ich für Disketten-Stationsbesitzer, folgende Änderungen vorgenommen:
505 *DISPLAY AT (2,1) BEEP: "WELCHES VOKABELPROGRAMM?"*
506 *ACCEPT AT (3,1)VALIDATE(UALPHA): WO§* **510** *OPEN #1: "DSK1."&WO§, OUTPUT, INTERNAL, FIXED 64*
§ = Dollarzeichen, # = Nummerzeichen
WO§ ermöglicht mir die eingegebenen 30 Vokabeln unter einem bestimmten Namen (bis zu 10 Buchstaben) abzuspeichern.

W. Bürger, Velbert

Betrifft: Programm Mampfmann aus Homecomputer Juli/83
Um Speicherplatz zu sparen, sollte man die Überschrift in eine Gosub-Routine bringen, ebenso die PAUSE 5000 mit Print AT 210; "Drücken Sie eine Taste".
Einfügung eines Textes zur Erleichterung eines Spieles:
201 *PRINT Mampfmann*
202 *PRINT,, ,, Geben Sie gleich die Tasten ein, mit denen Sie Ihren Mampfmann bewegen wollen!*
203 *PRINT AT 21,0; Drücken Sie eine Taste!: PAUSE 5000*
204 *Cls*
Zeile 9000 zeigt Fehler an: Abgeändert läuft es besser:
9000 *FOR a = 144 TO 157: FOR b = 0 TO 7: READ c: Poke USR CHR= (a) + b,c: NEXT b: NEXT a*

J. Grobbin Delmenhorst

Zuerst einmal ein riesiges Kompliment für Ihre beiden Zeitschriften "CPU" und "Homecomputer". Ich habe schon viele Computerzeitschriften ausprobiert, doch erst Ihre beiden Hefte haben mich richtig überzeugt.
Seit Dezember 1982 besitze ich einen TI-99/4A und habe jetzt dank Ihrer Programme endlich den Durchblick geschafft.
Im März dieses Jahres, erwarb ich ein Lehrprogramm für TI-Basic. In diesen Programmen wird viel Text verarbeitet. Nun versuche ich seit geraumer Zeit, ein Problem zu lösen: Normalerweise wird in einem laufenden Programm der Text "von unten in den Bildschirm" hinein gefahren. Wie gesagt, bei diesem Lernprogramm wird der Text jedoch als Ganzes auf einmal abgebildet.
Könnten Sie vielleicht meine Adresse veröffentlichen, so daß sich andere Leser, mit mir in Verbindung setzen können.

Alexander Opaschowski, Heilholzkamp 1, 2050 Börnsen

CATCH N'GOGO

für den TI-99/4A

Das Programm läuft auf einem TI-99/4A mit Joysticks. Es handelt sich um ein Spiel für 2 Teilnehmer, die je eine Spielfigur mit dem Joystick über den Bildschirm bewegen.

Spieler 1 kann doppelt so oft ziehen wie Spieler 2 und muß diesen einfangen. Dabei stehen bei jedem Spiel, dank RANDOMIZE, neue Hindernisse im Weg. Der Gejagte kann sich jedoch bei höchster Gefahr in ein Fragezeichen retten, was bewirkt, daß beide Spieler an irgendeiner Stelle des Bildschirms neu wieder auftauchen. Der Jäger kann diese Möglichkeit ebenfalls benutzen, wenn ihm der Abstand zu seinem Opfer zu groß erscheint. Doch nicht jedes Fragezeichen bringt die Rettung und schon garnicht dasselbe Fragezeichen zweimal. Die meisten verhalten sich wie gewöhnliche Hindernisse.

Nach dem Titelbild des Programms und nach der Frage nach dem Weitermachen am Schluß geht es durch Drücken auf den Feuerknopf weiter. Durch FCTN 4 (CLEAR) wird das Programm beendet. Durch einige Programmiertricks, wie den Gebrauch des Befehls DEF Funktion, konnte erreicht werden, daß das Programm eine für BASIC recht schnelle Grafik besitzt.

```
100 REM      ****************
110 REM      * CATCH N'GOGO *
120 REM      ****************
130 REM      *KNOBLOCH/WEBER*
140 REM      ****************
150 DEF W=L-Y/4
160 DEF V=M+X/4
170 DEF A=Z-Y/4
180 DEF B=S+X/4
190 DEF Q=(C=E)+(C=D)+1
200 DEF R=(C=F)+(C=D)+1
210 D=32
220 E=42
230 F=43
240 J=1
250 K=2
260 HI=32768
270 P=0
280 CALL CLEAR
290 CALL CHAR(E,"1898E725243CE404")
300 FOR I=2 TO 12
310 CALL CHAR(62,"0038440408100010")
320 CALL CHAR(64,"4949FF2424FF9494")
340 CALL COLOR(I,2,11)
350 NEXT I
360 CALL COLOR(1,2,4)
370 CALL SCREEN(6)
380 PRINT :::"       #################"
390 PRINT "       # CATCH N' GOGO #"
400 PRINT "       #################"::::
410 PRINT "    ";CHR$(E);" --- DER JAEGER":::
420 PRINT "    ";CHR$(F);" --- DER GEJAGTE":::
430 PRINT "    ";CHR$(64);" --- EIN HINDERNIS":::
440 PRINT "    ? --- DAS FRAGEZEICHEN            BINGT DIE RETTUNG";
445 PRINT "           ??? VIELLEICHT ???"::::
450 RESTORE
460 FOR I=1 TO 24
470 READ AA,BB,CC
480 CALL SOUND (AA,BB,CC)
490 CALL KEY(1,Z,S)
500 CALL KEY(2,Z,T)
510 IF S+T THEN 540
520 NEXT I
530 GOTO 450
540 CALL CLEAR
550 CALL COLOR(5,16,5)
560 PRINT ::::::::::::::::::::::::
```

```
570 CALL SCREEN(7)
580 CALL HCHAR(1,2,64,31)
590 CALL HCHAR(24,2,64,31)
600 CALL VCHAR(2,2,64,22)
610 CALL VCHAR(2,32,64,22)
620 CALL VCHAR(2,31,32,22)
630 CALL COLOR(1,4,4)
640 RANDOMIZE
650 FOR I=1 TO 100
660 CALL HCHAR(21*RND+2,28*RND+3,64)
670 NEXT I
680 CALL COLOR(4,4,4)
690 FOR I=1 TO 25
700 CALL HCHAR(21*RND+2,28*RND+3,62)
710 NEXT I
720 FOR I=1 TO 25
730 CALL HCHAR(21*RND+2,28*RND+3,63)
740 NEXT I
750 CALL COLOR(4,16,14)
760 CALL COLOR(2,2,4)
770 Z=21*RND+2
780 S=28*RND+3
790 L=21*RND+2
800 M=28*RND+3
810 CALL HCHAR(Z,S,E)
820 CALL HCHAR(L,M,F)
830 CALL JOYST(J,X,Y)
840 CALL GCHAR(A,B,C)
850 IF Q THEN 1560
860 CALL HCHAR(Z,S,D)
870 S=B
880 Z=A
890 CALL HCHAR(Z,S,E)
900 CALL JOYST(K,X,Y)
910 CALL GCHAR(W,V,C)
920 IF R THEN 1060
930 CALL HCHAR(L,M,D)
940 L=W
950 M=V
960 CALL HCHAR(L,M,F)
970 CALL JOYST(J,X,Y)
980 CALL GCHAR(A,B,C)
990 IF Q THEN 1560
1000 CALL HCHAR(Z,S,D)
1010 S=B
1020 Z=A
1030 CALL HCHAR(Z,S,E)
1040 P=P+J
1050 GOTO 830
1060 IF C<63 THEN 1090
1070 CALL SOUND(-20,440,0)
1080 GOTO 830
1090 IF C<>62 THEN 1150
1100 CALL SOUND(-500,524,0)
1110 CALL HCHAR(W,V,63)
1120 CALL HCHAR(Z,S,D)
1130 CALL HCHAR (L,M,D)
1140 GOTO 770
1150 CALL HCHAR(Z,S,D)
1160 CALL COLOR(8,16,7)
1170 CALL HCHAR(L,M,88)
1180 CALL SOUND(50,262,0,330,1,392,2)
1190 CALL SCREEN(11)
1200 CALL CLEAR
1210 FOR I=2 TO 12
1220 CALL COLOR(I,2,1)
```

```
1230 NEXT I
1240 IF P<HI THEN 1380
1250 PRINT :::::::::"GESCHAFFT NACH";P;"RUNDEN"::::"BESTLEISTUNG WAR";
1255 PRINT HI:::::"WOLLEN SIE WIRKLICH NOCHMAL?"::::::::
1260 CALL SOUND(1000,262,0,330,1,392,2)
1270 CALL SOUND(1000,330,0,392,1,524,2)
1280 CALL SOUND(2000,392,0,524,1,660,2)
1290 RESTORE
1300 FOR I=1 TO 24
1310 READ AA,BB,CC
1320 CALL SOUND(AA,BB,CC)
1330 CALL KEY(1,Z,S)
1340 CALL KEY(2,Z,T)
1350 IF S+T THEN 270
1360 NEXT I
1370 GOTO 1290
1380 HI=P
1390 PRINT :::::::::"GESCHAFFT NACH";P;"RUNDEN"::::"NEUE BESTLEISTUNG"
1395 PRINT ::::"WOLLEN SIE WIRKLICH NOCHMAL?"::::::::
1400 CALL SOUND(1000,262,0,330,1,392,2)
1410 FOR I=1 TO 27
1420 K=-K
1430 CALL SCREEN(9-K)
1440 NEXT I
1450 CALL SOUND(1000,330,0,392,1,524,2)
1460 FOR I=1 TO 27
1470 K=-K
1480 CALL SCREEN(9-K)
1490 NEXT I
1500 CALL SOUND(2000,392,0,524,1,660,2)
1510 FOR I=1 TO 54
1520 K=-K
1530 CALL SCREEN(9-K)
1540 NEXT I
1550 GOTO 1290
1560 IF C<63 THEN 1090
1570 CALL SOUND(-20,440,0)
1580 GOTO 900
1590 IF C<>62 THEN 1150
1600 CALL SOUND(-500,524,0)
1610 CALL HCHAR(A,B,63)
1620 CALL HCHAR(Z,S,D)
1630 CALL HCHAR(L,M,D)
1640 GOTO 770
1650 DATA 300,349,0
1660 DATA 300,440,0
1670 DATA 300,524,0
1680 DATA 600,392,0
1690 DATA 25,262,0
1700 DATA 25,262,0
1710 DATA 50,262,0
1720 DATA 300,110,30
1730 DATA 300,349,0
1740 DATA 300,440,0
1750 DATA 300,524,0
1760 DATA 600,392,0
1770 DATA 25,262,0
1780 DATA 25,262,0
1790 DATA 50,262,0
1800 DATA 300,110,30
1810 DATA 300,392,0
1820 DATA 300,349,0
1830 DATA 300,294,0
1840 DATA 600,262,0
1850 DATA 25,262,0
1860 DATA 25,262,0
1870 DATA 50,262,0
1880 DATA 50,110,30
```

Raumschiff Enterprise

für den TI-99/4A

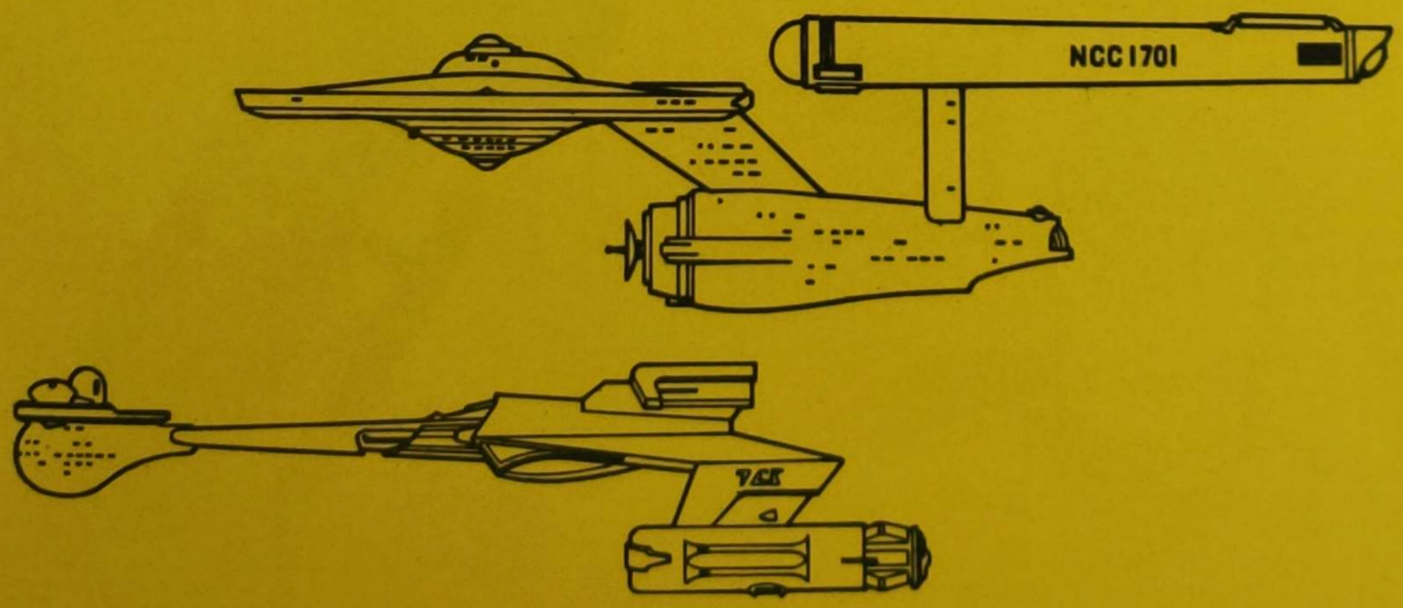
Ziel wares, ein TI-Basic-Programm zuentwickeln, dasprofessionell aufgebaut, Schnelligkeit und anspruchsvolle Grafik miteinander verbindet. Dieses Programm ist jedem TI-99/4A Besitzer zugänglich, und zeigt andererseits vielen Kritikern, daß nicht unbedingt Extended Basic notwendig ist, wenn man nur die Möglichkeiten von TI-Basic erkennt und ausnutzt. - Das Spiel stützt sich auf die bekannte Serie "Raumschiff Enterprise".

Sie müssen feindliche Klingonenschiffe vom Beschuß zweier unbewaffneter Raumstationen abhalten. Für jeden Treffer erhalten Sie 100 Punkte. Wird aber eine der Raumstationen getroffen, werden Ihnen 100 Punkte abgezogen. Gelegentlich tauchen zwei Klingonenschiffe unterhalb der Raumstationen auf. Sinken Sie in diesem Fall sofort (Fernbedienung einfach nach unten) und setzen Sie Ihre Laserkanonen ein. Wenn Sie wieder steigen, beginnt die nächste Runde. (Wenn Sie sich während eines Spiels ausruhen wollen, können Sie das an dieser Stelle tun. Erst wenn Sie gestiegen sind, erscheinen neue Klingonen. Drücken Sie einfach die Alpha-Lock-Taste, und kein "Unbefugter" kann für Sie weiterspielen).

Wenn Ihnen die Energie für Ihre Photonentorpedos und Laser ausgegangen ist, bewertet der Computer Ihre Leistung. Übrigens: Keine Angst! Inzwischen können sich die Raumstationen, dank eines Umbaues, selbst verteidigen. Falls dieses System einmal ausfallen sollte, starten Sie einfach erneut das Programm und übernehmen Sie die Verteidung.

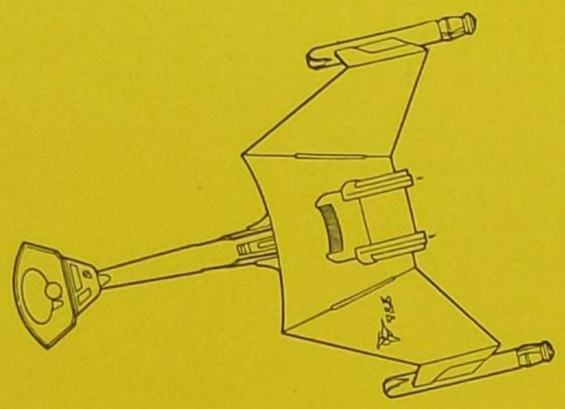
100- 360 Zeichendefinitionen
370- 480 Farbverordnungen
490- 610 A$ waagrecht oder senkrecht zeichnen.
620- 680 Waagerechter Enterprise-Laser
690- 760 Enterprise zeichnen
770-1510 Titelbild zeichnen (Enterprise bewegen und Abfrage, Erläuterungen, Spielbeginn).
1520-1750 Erläuterungen
1760-1850 Schwierigkeitswahl
1860-2130 Zeichendefinitionen
2140-2430 Zeichnen (Sterne, Raumstationen)
2440-2520 Farbzuordnungen
2530-3310 eigentliches Spiel (Abfrage, Fernbedienung, Firetaste, senkrechter Laser)
3320-3570 Klingonen (neues Schiff erscheinen lassen oder:)
3580-3920 Klingonen-Laser
3930-3980 Klingonenschiff zeichnen
3990-4750 Kingonenschiff unter Raumstationen
4760-4880 Enterprise-Fanfare
4890-5120 Ende

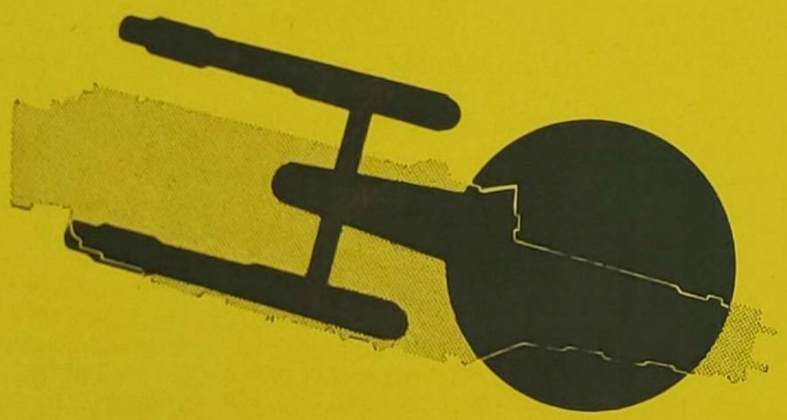
Die Variablen Z,Z1,Z2,Z3,Z0 werden in FOR..NEXT-Schleifen und ähnlichen, universellen Anwendungen benutzt. Die Variablen-Spalte, Zeile, Länge erklären sich selbst. Auf REM-Zeilen und der Möglichkeit, das Programm wieder von vorne zu starten (alle Variablen müssen = 0 gesetzt werden), wurde aus Speicherplatzgründen zugunsten vieler Details verzichtet. (Versuchen Sie ruhig einmal, auf die Raumstationen zu schießen, oder schießen, wenn kein Klingonenschiff da ist,der Computer ignoriert dies nicht (wie üblich)!



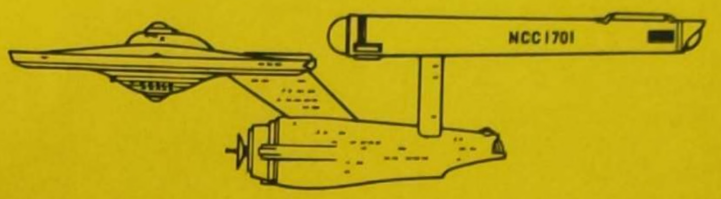
```
100 REM === RAUMSCHIFF ===          === ENTERPRISE ===          ==TEXAS TI-99/4A
==    =1983===PATRICK SCHMITZ=          548 REM.-UNKELBACH
110 CALL CLEAR
120 CALL SCREEN(2)
130 CALL CHAR(64,"384492A2A292443B")
140 CALL CHAR(95,"")
150 CALL CHAR(39,"FF")
160 CALL CHAR(33,"00202070702")
170 DATA 00FFFFFF0E0E0E0E,0E0E0E0E0E0E0E0E,0E0E0E0E0E0E0E0E,00E7E7E707070707,070
7070707070707,0707070707070707
180 DATA 0001070707070707,0707017070000000,0000000000070707,00F0FCFC1C1C1C1C,FCF
CFC1C1C1C1C1C,1C1C1C1C1CFCFCFC
190 DATA 003EFFFFE3E3E3E3,FFFF1F0303030303,0303030303FFFFFF,0000808080808181,818
383838686868C,8D8D999898B0B0B0
200 DATA 006F6FCFDBDB9BB3,B333636363C3C3C3,FFFFFF0303030303,0081818183838383,878
787878E8E8E8E,DFDFDF9FB8B8B8B8
210 DATA 00F8F8F89C9C9C9C,0E0E0E0E07070707,FFFFFFFF01010101,0000000000000000,000
0000000000000,B0B0B0B0C0C0C0C0
220 FOR Z=128 TO 157
230 READ Z$
240 CALL CHAR(Z,Z$)
250 NEXT Z
260 B$(1)="FF7F"
270 B$(2)="FCFE402010080402"
280 B$(3)="00017FFF7F214080"
290 B$(4)="0080FEFFFE8"
300 B$(5)="070F"
310 B$(6)="E0F"
320 FOR Z1=95 TO 119 STEP 8
330 FOR Z=1 TO 6
340 CALL CHAR(Z1+Z,B$(Z))
350 NEXT Z
360 NEXT Z1
370 CALL COLOR(1,12,2)
380 CALL COLOR(2,16,5)
390 FOR Z=3 TO 8
400 CALL COLOR(Z,16,7)
410 NEXT Z
420 FOR Z=9 TO 12
430 CALL COLOR(Z,2,2)
440 NEXT Z
450 FOR Z=13 TO 16
460 CALL COLOR(Z,9,2)
470 NEXT Z
480 GOTO 770
490 ZEILE=VAL(SEG$(A$,2,2))
500 SPALTE=VAL(SEG$(A$,4,2))
510 IF SEG$(A$,1,1)="H" THEN 570
520 FOR Z=6 TO LEN(A$)
530 CALL HCHAR(ZEILE,SPALTE,ASC(SEG$(A$,Z,Z)))
540 ZEILE=ZEILE+1
550 NEXT Z
560 RETURN
570 FOR Z=6 TO LEN(A$)
580 CALL HCHAR(ZEILE,SPALTE,ASC(SEG$(A$,Z,1)))
590 SPALTE=SPALTE+1
600 NEXT Z
610 RETURN
620 LAENGE=19
630 CALL HCHAR(ZEILE,9,39,LAENGE)
640 FOR Z=0 TO 7
650 CALL SOUND(-100,1000,Z,990,Z,1010,Z)
660 NEXT Z
670 CALL HCHAR(ZEILE,9,32,LAENGE)
680 RETURN
690 FOR Z=SPALTE TO SPALTE+3
700 CALL HCHAR(ZEILE,Z,ZEICHEN+Z0)
```

```
710 Z0=Z0+1
720 NEXT Z
730 CALL HCHAR(ZEILE+1,SPALTE+1,ZEICHEN+4)
740 CALL HCHAR(ZEILE+1,SPALTE+2,ZEICHEN+5)
750 Z0=0
760 RETURN
770 FOR Z=1 TO 7
780 CALL HCHAR(Z,3,42,28)
790 NEXT Z
800 FOR Z=22 TO 24
810 CALL HCHAR(Z,3,42,28)
820 NEXT Z
830 FOR Z=3 TO 28 STEP 25
840 FOR Z1=Z TO Z+2
850 CALL VCHAR(1,Z1,42,24)
860 NEXT Z1
870 NEXT Z
880 FOR Z=12 TO 21
890 FOR Z1=3 TO 5
900 CALL HCHAR(Z1,Z,89+Z1+Z*3)
910 NEXT Z1
920 NEXT Z
930 CALL HCHAR(4,4,95,8)
940 A$="V0404_TEXAS_INSTRUMENTS_"
950 GOSUB 490
960 CALL HCHAR(4,22,95,7)
970 A$="V0429_TEXAS_INSTRUMENTS_"
980 GOSUB 490
990 A$="H2304__§_1983_PATRICK_SCHMITZ__"
1000 GOSUB 490
1010 A$="H0906_RAUMSCHIFF_"
1020 GOSUB 490
1030 A$="H1016_ENTERPRISE_"
1040 GOSUB 490
1050 FOR ZEICHEN=96 TO 120 STEP 8
1060 ZEILE=12
1070 SPALTE=ZEICHEN/2-42
1080 GOSUB 690
1090 NEXT ZEICHEN
1100 A$="H1510 1>ERLAEUTERUNGEN"
1110 GOSUB 490
1120 A$="H1710 2>SPIELBEGINN___"
1130 GOSUB 490
1140 CALL COLOR(9,16,2)
1150 GOSUB 4760
1160 CALL SOUND(100,40000,30)
1170 GOSUB 4810
1180 FOR Z=10 TO 12
1190 CALL COLOR(Z-1,2,2)
1200 CALL COLOR(Z,16,2)
1210 GOSUB 1290
1220 NEXT Z
1230 FOR Z=11 TO 9 STEP -1
1240 CALL COLOR(Z+1,2,2)
1250 CALL COLOR(Z,16,2)
1260 GOSUB 1290
1270 NEXT Z
1280 IF Z2=10 THEN 1360 ELSE 1180
1290 IF Z2=10 THEN 1350
1300 CALL KEY(0,KEY,STATUS)
1310 IF (KEY=49)+(KEY=50)=-1 THEN 1320 ELSE 1350
1320 Z2=10
1330 CALL SOUND(-50,330,4,220,4)
1340 CALL SOUND(-50,330,4,440,4)
1350 RETURN
1360 FOR ZEICHEN=104 TO 112 STEP 8
1370 SPALTE=6
```

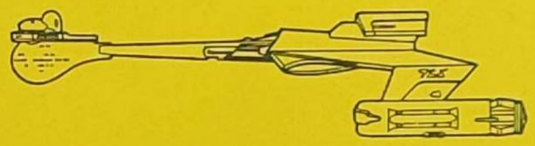


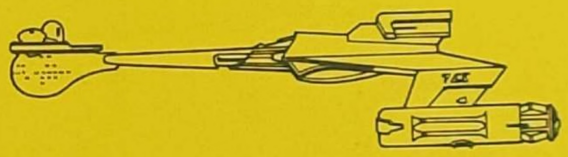


```
1380 ZEILE=ZEICHEN/4-12
1390 GOSUB 690
1400 NEXT ZEICHEN
1410 CALL HCHAR(12,10,32,15)
1420 CALL HCHAR(13,10,32,15)
1430 CALL COLOR(9,2,2)
1440 CALL COLOR(10,16,2)
1450 ZEILE=15
1460 GOSUB 620
1470 CALL COLOR(10,2,2)
1480 CALL COLOR(11,16,2)
1490 ZEILE=17
1500 GOSUB 620
1510 IF KEY=50 THEN 1760
1520 CALL CLEAR
1530 FOR Z=2 TO 8
1540 CALL COLOR(Z,16,2)
1550 NEXT Z
1560 PRINT "DER WELTRAUM -": :" UNENDLICHE WEITEN.": :"WIR SCHREIBEN DAS JAHR 22
00.": :"DIES SIND DIE ABENTEUER DES"
1570 PRINT :"RAUMSCHIFFS  ENTERPRISE ,": :"DAS MIT SIENER 400 MANN": :"STARKEN B
ESATZUNG 5 JAHRE": :
1580 PRINT "LANG UNTERWEGS IST, UM": :"NEUE WELTEN ZU ERFORSCHEN,": :"NEUES LEBE
N UND": :"NEUE ZIVILISATION.":TAB(26);">>"
1590 GOSUB 4760
1600 GOSUB 1720
1610 PRINT "VIELE LICHTJAHRE VON DER": :"ERDE ENTFERNT DRINGT DIE": :"ENTERPRISE
  IN GALAXIEN VOR,": :
1620 PRINT "DIE NIE EIN MENSCH ZUVOR": :"GESEHEN HAT. ": : : :TAB(26);">>"
1630 GOSUB 4810
1640 GOSUB 1720
1650 PRINT "**SPIELREGELN**": :"FEINDLICHE KLINGONSCHIFFE  VERSUCHEN ZWEI FRIED
LICHE  FORSCHUNGS-RAUMSTATIONEN DER FOEDERATION DER VEREINIGTENPLANETEN  ZU ZER
STOEREN."
1660 PRINT "IHR AUFTRAG IST ES, DIES MITDER  ENTERPRISE  ZU VERHIN- DERN. WENN S
IE IHRE PHOTONENTORPEDOS NICHT RECHTZEITIG  EINSETZEN, BEGINNEN DIE "
1670 PRINT "KLINGONEN DEN BESCHUSS DER  RAUMSTATIONEN. GELEGENTLICH ERSCHEINEN Z
WEI KLINGON-":"SCHIFFE UNTERHALB DER RAUM- STATIONEN. SINKEN SIE IN"
1680 PRINT "DIESEM FALL SOFORT UND ZER- STOEREN SIE DIE KLINGON-":"SCHIFFE MIT E
INEM LASER-":"STRAHL. NACHDEM SIE WIEDER  GESTIEGEN SIND, BEGINNT DIE"
1690 PRINT "NAECHSTE RUNDE.";TAB(27);">>"
1700 GOSUB 1720
1710 GOTO 1810
1720 CALL KEY(0,KEY,STATUS)
1730 IF STATUS=0 THEN 1720
1740 CALL CLEAR
1750 RETURN
1760 A$="H1710 VIEL_ERFOLG"
1770 GOSUB 490
1780 FOR Z=1 TO 150
1790 NEXT Z
1800 GOSUB 620
1810 CALL CLEAR
1820 PRINT "     SCHWIERIGKEITSGRAD:": : :TAB(8);"1>  EINFACH": :TAB(8);"2> E S
CHWER": :TAB(8);"3> SUPERSCHWER": : : : : : :
1830 CALL KEY(0,LE,STATUS)
1840 IF (LE=49)+(LE=50)+(LE=51)=0 THEN 1830
1850 CALL SOUND(100,220,0,247,0,262,0)
1860 CALL CLEAR
1870 FOR Z1=127 TO 135 STEP 8
1880 FOR Z=1 TO 6
1890 CALL CHAR(Z1+Z,B$(Z))
1900 NEXT Z
1910 NEXT Z1
1920 RESTORE 1930
1930 DATA 00707070180C0603,000000183C7EFFBD,000E0E0E183060C,03060C1830303,3C3C18
,C06030180C0C0C
```



```
1940 FOR Z=1 TO 6
1950 READ B$(Z)
1960 CALL CHAR(Z+87,B$(Z))
1970 NEXT Z
1980 RESTORE 1990
1990 DATA 0000000010383C3E,0,0,0,3B11,0080C06E3F796C67,0018183CFCF87CFE,0,000000
0000000001,33373D1F3767C78F
2000 DATA F7F3F9CCC683E37F,0080C0E06060C0C0,23767C78702,0F0606,1C,62371F0F0702
2010 FOR Z=144 TO 159
2020 READ Z$
2030 CALL CHAR(Z,Z$)
2040 NEXT Z
2050 CALL CHAR(58,"FFFFF7E3F7FFFFFF")
2060 CALL CHAR(59,"FFFFEFF7FFFFFFFF")
2070 CALL CHAR(60,"FFFFE7C3E7FFFFFF")
2080 CALL CHAR(61,"FFFFF7E3E3F7FFFF")
2090 CALL CHAR(64,"")
2100 CALL CHAR(94,"04104004209401")
2110 CALL CHAR(95,"A8412448B22C485C")
2120 CALL CHAR(62,"DD77D5BD4DF7BF76")
2130 CALL CHAR(38,"1818181818181818")
2140 FOR Z=2 TO 16
2150 CALL COLOR(Z,2,2)
2160 NEXT Z
2170 RANDOMIZE
2180 FOR Z=1 TO 5*RND+50
2190 CALL HCHAR(INT(18*RND)+4,INT(25*RND)+5,INT(4*RND)+58)
2200 NEXT Z
2210 CALL VCHAR(1,3,42,24)
2220 CALL VCHAR(1,30,42,24)
2230 CALL HCHAR(1,3,42,28)
2240 CALL HCHAR(3,3,42,28)
2250 FOR Z=21 TO 24
2260 CALL HCHAR(Z,3,42,28)
2270 NEXT Z
2280 A$="H0203***RAUMSCHIFF§§ENTERPRISE***"
2290 GOSUB 490
2300 A$="H2206RUNDE***PUNKTE***LASER"
```





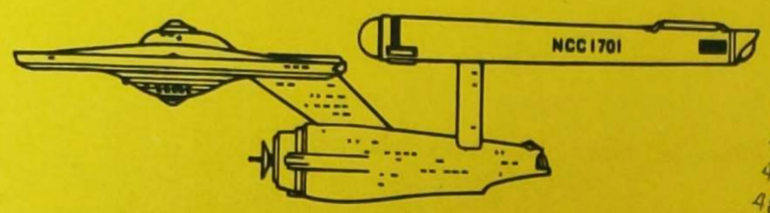
```
2310 GOSUB 490
2320 ZEILE=5
2330 FOR ZEICHEN=96 TO 136 STEP 8
2340 SPALTE=ZEICHEN/2-44
2350 GOSUB 690
2360 NEXT ZEICHEN
2370 FOR Z2=12 TO 16 STEP 4
2380 FOR Z1=1 TO 4
2390 FOR Z=1 TO 4
2400 CALL HCHAR(9+Z2/4+Z1,Z2+Z,139+Z+Z1*4)
2410 NEXT Z
2420 NEXT Z1
2430 NEXT Z2
2440 CALL COLOR(2,16,5)
2450 CALL COLOR(3,2,15)
2460 CALL COLOR(4,2,15)
2470 FOR Z=5 TO 7
2480 CALL COLOR(Z,16,7)
2490 NEXT Z
2500 CALL COLOR(8,4,2)
2510 CALL COLOR(15,8,2)
2520 CALL COLOR(16,8,2)
2530 RUNDE=1
2540 A$="H23081******000*******50"
2550 GOSUB 490
2560 LASER=50
2570 EPOS=9
2580 CALL COLOR(9,16,2)
2590 Z0=0
```

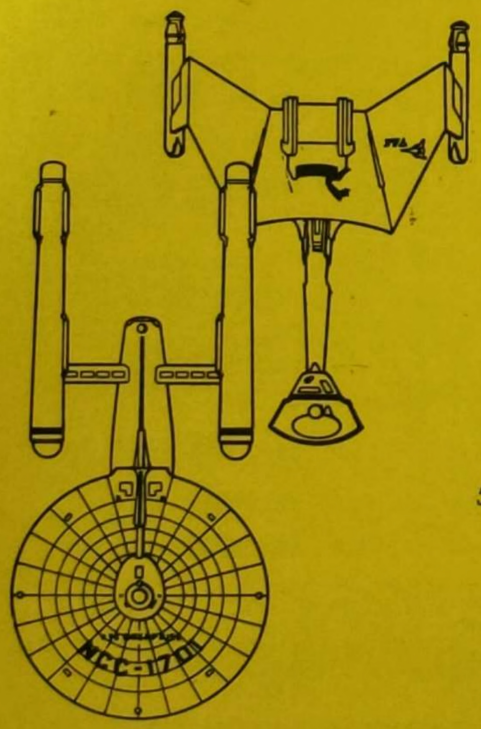
```
2600 CALL JOYST(1,X,Y)
2610 IF X+Y=0 THEN 2670
2620 IF (EPOS+X/4<15)+(EPOS+X/4>8)<>-2 THEN 2660
2630 CALL COLOR(EPOS,2,2)
2640 EPOS=EPOS+X/4
2650 CALL COLOR(EPOS,16,2)
2660 IF RND<LE/10-4.8 THEN 3320
2670 CALL KEY(1,KEY,STATUS)
2680 IF KEY=18 THEN 2710
2690 IF RND<LE/10-4.8 THEN 3320
2700 GOTO 2600
2710 LASER=LASER-1
2720 A$="H2325"&STR$(LASER)
2730 GOSUB 490
2740 IF LASER<>9 THEN 2760
2750 CALL HCHAR(23,26,42)
2760 IF LASER=0 THEN 4890
2770 Z=1
2780 IF EPOS*4-30<>KPOS(Z)THEN 2830
2790 SPALTE=KPOS(Z)+1
2800 LAENGE=3
2810 KPOS(Z)=0
2820 GOTO 3050
2830 IF Z=2 THEN 2860
2840 Z=2
2850 GOTO 2780
2860 Z=1
2870 IF EPOS*4-30<>KPOS1(Z)THEN 2950
2880 SPALTE=KPOS1(Z)+1
2890 IF (SPALTE=15)+(SPALTE=19)=-1 THEN 2920
2900 LAENGE=8
2910 GOTO 2930
2920 LAENGE=5
2930 KPOS1(Z)=0
2940 GOTO 3050
2950 IF Z=2 THEN 2980
2960 Z=2
2970 GOTO 2870
2980 SPALTE=EPOS*4-29
2990 IF (SPALTE=15)+(SPALTE=19)<>-1 THEN 3030
3000 LAENGE=8
3010 Z3=9
3020 GOTO 3050
3030 LAENGE=15
3040 Z3=10
3050 CALL VCHAR(6,SPALTE,33,LAENGE)
3060 FOR Z1=0 TO 8 STEP 2.5
3070 CALL SOUND(-100,1000,Z1,990,Z1,1010,Z1)
3080 NEXT Z1
3090 CALL VCHAR(6,SPALTE,32,LAENGE)
3100 IF Z3=10 THEN 3300
3110 IF Z3<>9 THEN 3170
3120 CALL HCHAR(LAENGE+SPALTE/2.6,SPALTE,62)
3130 CALL SOUND(25,110,0,120,0,130,0)
3140 CALL SOUND(-300,-7,0)
3150 CALL HCHAR(LAENGE+SPALTE/2.6,SPALTE,150)
3160 GOTO 3860
3170 CALL VCHAR(LAENGE+6,SPALTE,95,2)
3180 CALL SOUND(-500,-5,0)
3190 CALL VCHAR(LAENGE+6,SPALTE,32,2)
3200 FOR Z0=94 TO 32 STEP -62
3210 FOR Z1=SPALTE-1 TO SPALTE+1 STEP 2
3220 CALL VCHAR(LAENGE+6,Z1,Z0,2)
3230 CALL SOUND(-200,(SPALTE-Z1=1)-6,0)
3240 NEXT Z1
3250 NEXT Z0
3260 IF UNTEN<>0 THEN 4340
```

```
3270 PUNKTE=PUNKTE+100
3280 A$="H2315"&STR$(PUNKTE)
3290 GOSUB 490
3300 Z3=0
3310 GOTO 2600
3320 IF KPOS1(1)<>0 THEN 3580
3330 IF KPOS1(2)<>0 THEN 3600
3340 FOR Z=1 TO ZAHL
3350 IF KPOS(Z)=0 THEN 3460
3360 IF (KPOS(Z)=14)+(KPOS(Z)=18)=-1 THEN 3370 ELSE 3390
3370 Z0=11
3380 GOTO 3400
3390 Z0=14
3400 CALL HCHAR(9,KPOS(Z),32,3)
3410 CALL HCHAR(10,KPOS(Z),32,3)
3420 GOSUB 3930
3430 KPOS1(Z)=KPOS(Z)
3440 KPOS(Z)=0
3450 Z0=100
3460 NEXT Z
3470 IF Z0=100 THEN 2590
3480 IF RN>4 THEN 3990
3490 RN=RN+1
3500 ZAHL=INT(2*RND+1)
3510 FOR Z=1 TO ZAHL
3520 KPOS(Z)=INT(6*RND+1)*4+2
3530 IF KPOS(1)=KPOS(2)THEN 3520
3540 Z0=9
3550 GOSUB 3930
3560 NEXT Z
3570 GOTO 2600
3580 Z=1
3590 GOTO 3610
3600 Z=2
3610 CALL COLOR(1,10,2)
3620 ON KPOS1(Z)/4-.5 GOTO 3630,3650,3670,3670,3750,3770
3630 LASER$="16090320415613"
3640 GOTO 3780
3650 LASER$="15131520115314"
3660 GOTO 3780
3670 CALL VCHAR(13,13+(KPOS1(Z)/4-3)*4,38,2)
3680 CALL SOUND(-50,-3,0)
3690 CALL SOUND(-50,-3,5)
3700 CALL VCHAR(13,13+(KPOS1(Z)/4-3)*4,32,2)
3710 CALL HCHAR(10+KPOS1(Z)/4,13+(KPOS1(Z)/4-3)*4,62)
3720 CALL SOUND(-500,110,30,110,30,5000,30,-8,0)
3730 CALL HCHAR(10+KPOS1(Z)/4,13+(KPOS1(Z)/4-3)*4,150)
3740 GOTO 3850
3750 LASER$="16210320115520"
3760 GOTO 3780
3770 LASER$="16200320615520"
3780 CALL HCHAR(VAL(SEG$(LASER$,1,2)),VAL(SEG$(LASER$,3,2)),39,VAL(SEG$(LASER$,8
,2)))
3790 CALL SOUND(-50,-3,0)
3800 CALL SOUND(-50,-3,5)
3810 CALL HCHAR(VAL(SEG$(LASER$,1,2)),VAL(SEG$(LASER$,3,2)),VAL(SEG$(LASER$,5,3)
),VAL(SEG$(LASER$,8,2)))
3820 CALL HCHAR(VAL(SEG$(LASER$,1,2)),VAL(SEG$(LASER$,13,2)),62)
3830 CALL SOUND(-500,110,30,110,30,5000,30,-8,0)
3840 CALL HCHAR(VAL(SEG$(LASER$,1,2)),VAL(SEG$(LASER$,13,2)),VAL(SEG$(LASER$,10,
3)))
3850 CALL COLOR(1,12,2)
3860 PUNKTE=PUNKTE-200
3870 IF PUNKTE<0 THEN 3890
3880 IF PUNKTE>1000 THEN 3910 ELSE 3900
3890 PUNKTE=0
3900 CALL HCHAR(23,18,42)
```

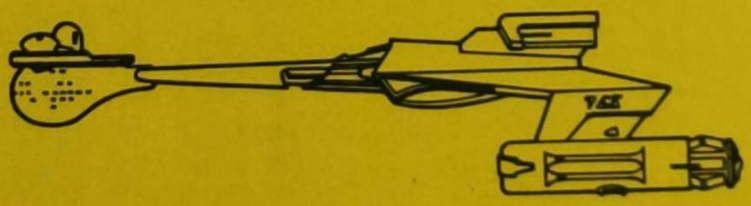
```
3910 GOTO 3280
3920 GOTO 2600
3930 FOR Z1=Z0 TO Z0+1
3940 FOR Z2=KPOS(Z)TO KPOS(Z)+2
3950 CALL HCHAR(Z1,Z2,88+(Z1-Z0)*3+Z2-KPOS(Z))
3960 NEXT Z2
3970 NEXT Z1
3980 RETURN
3990 RN=0
4000 FOR Z1=15 TO 18 STEP 3
4010 FOR Z2=19 TO 20
4020 FOR Z3=Z1 TO Z1+2
4030 CALL HCHAR(Z2,Z3,31+3*Z2+Z3-Z1)
4040 NEXT Z3
4050 NEXT Z2
4060 NEXT Z1
4070 CALL JOYST(1,X,Y)
4080 IF (X=0)+(Y=-4)=-2 THEN 4090 ELSE 4160
4090 CALL HCHAR(5,EPOS*4-32,32,4)
4100 CALL HCHAR(6,EPOS*4-32,32,4)
4110 Z0=0
4120 ZEICHEN=EPOS*8+24
4130 ZEILE=19
4140 SPALTE=6
4150 GOSUB 690
4160 CALL KEY(1,KEY,STATUS)
4170 IF (KEY=18)+(Z0=0)<>-2 THEN 4380
4180 LASER=LASER-1
4190 IF LASER<>9 THEN 4210
4200 CALL HCHAR(23,26,42)
4210 A$="H2325"&STR$(LASER)
4220 GOSUB 490
4230 IF LASER=0 THEN 4890
4240 ZEILE=20
4250 LAENGE=12
4260 IF Z00<>1 THEN 4280
4270 LAENGE=21
4280 GOSUB 630
4290 IF Z00=1 THEN 4620
4300 UNTEN=100
4310 LAENGE=13
4320 SPALTE=16
4330 GOTO 3170
4340 IF UNTEN=50 THEN 4580
```

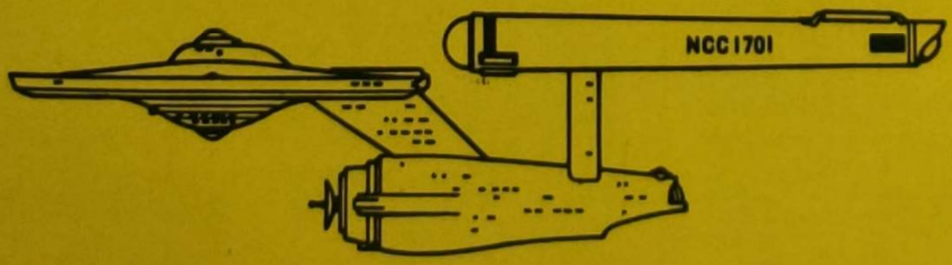
```
4350 UNTEN=50
4360 SPALTE=19
4370 GOTO 3170
4380 IF Z00=1 THEN 4620
4390 IF RND<.5 THEN 4070
4400 PUNKTE=PUNKTE-200
4410 IF PUNKTE<0 THEN 4430
4420 IF PUNKTE>1000 THEN 4450 ELSE 4440
4430 PUNKTE=0
4440 CALL HCHAR(23,18,42)
4450 A$="H2315"&STR$(PUNKTE)
4460 GOSUB 490
4470 CALL COLOR(1,10,2)
4480 CALL VCHAR(17,19,38,2)
4490 FOR Z1=0 TO 6 STEP 2
4500 CALL SOUND(-100,110,Z1)
4510 NEXT Z1
4520 CALL VCHAR(17,19,32,2)
4530 CALL HCHAR(17,19,62)
4540 CALL SOUND(-500,110,30,110,30,5500,30,-8,0)
4550 CALL HCHAR(17,19,158)
4560 CALL COLOR(1,12,2)
4570 GOTO 4070
4580 UNTEN=0
4590 RUNDE=RUNDE+1
4600 A$="H2308"&STR$(RUNDE)
4610 GOSUB 490
4620 Z00=1
4630 Z0=0
4640 CALL JOYST(1,X,Y)
4650 IF (X=0)+(Y=4)=-2 THEN 4660 ELSE 4160
4660 CALL HCHAR(19,6,32,4)
4670 CALL HCHAR(20,6,32,4)
4680 Z0=0
4690 ZEILE=5
4700 SPALTE=EPOS*4-32
4710 ZEICHEN=EPOS*8+24
4720 GOSUB 690
4730 IF LASER=0 THEN 4900
4740 Z00=0
4750 GOTO 2600
```






```
4760 CALL SOUND(300,523,6,1047,0,262,8)
4770 CALL SOUND(200,466,6,932,0,233,8)
4780 CALL SOUND(200,440,6,880,0,220,8)
4790 CALL SOUND(300,392,6,784,0,196,8)
4800 RETURN
4810 CALL SOUND(250,349,6,698,0,175,8)
4820 CALL SOUND(200,392,6,784,0,196,8)
4830 CALL SOUND(200,392,6,784,0,196,8)
4840 CALL SOUND(200,440,6,880,0,220,8)
4850 CALL SOUND(300,466,6,932,0,233,8)
4860 CALL SOUND(200,466,6,932,0,233,8)
4870 CALL SOUND(250,523,6,1047,0,262,8)
4880 RETURN
4890 GOTO 4660
4900 A$="H1904§SIE§HABEN§IHREN§AUFTRAG§§"
4910 GOSUB 490
4920 IF PUNKTE>3500 THEN 4980
4930 IF PUNKTE>2500 THEN 4960
4940 A$="NICHT§SCHLECHT§"
4950 GOTO 4990
4960 A$="§BESONDERS§GUT§"
4970 GOTO 4990
4980 A$="§§HERVORRAGEND§"
4990 A$="H2004"&A$&"AUSGEFUEHRT"
5000 GOSUB 490
5010 GOSUB 4760
5020 CALL SOUND(100,110,30)
5030 GOSUB 4810
5040 FOR Z1=10 TO 14
5050 CALL COLOR(Z1-1,2,2)
5060 CALL COLOR(Z1,16,2)
5070 NEXT Z1
5080 FOR Z1=13 TO 9 STEP -1
5090 CALL COLOR(Z1+1,2,2)
5100 CALL COLOR(Z1,16,2)
5110 NEXT Z1
5120 GOTO 5040
```





U-Boot ist ein Programm für einen Spieler, für den Ti-99/4A Homecomputer mit dem Extended-Basic-Modul. Nach dem Start mit RUN erscheint das Titelbild mit der Frage, ob sie eine Spielanleitung wünschen, wenn Sie die Frage bejahen, erhalten Sie die Spielanleitung auf dem Bildschirm.

Spielablauf:
Als Kommandant eines U-Bootes, ist es Ihre Aufgabe, so viel Schiffe wie möglich abzuschießen. Dies ist jedoch nicht so leicht wie man denken könnte, denn im Meer gibt es eine Menge Seetiere.
In jeder Runde stehen Ihnen 20 Torpedos als Munition zur Verfügung. Mit diesen Torpedos müssen Sie eine bestimmte Anzahl von Schiffen treffen um in die nächste Runde zu gelangen. Diese Mindestanzahl wird während der ganzen Runde in Klammern angezeigt und erhöht sich mit jeder Runde um 2 Treffer (bis max. 16).
Gelingt es Ihnen nicht, diese Mindestanzahl an Treffern zu erzielen, so ist das Spiel zu Ende und das Schlußbild erscheint auf dem Bildschirm.
Jedesmal, wenn Sie im Verlaufe des Spieles das Aloha Oe hören, müssen Sie irgendeine Taste drücken um die nächste Runde, bzw. um ein neues Spiel zu beginnen.

Hindernisse:
Wenn Sie mit Ihrem Torpedo ein Seepferd treffen, so werden Ihnen von Ihrem Punktekonto 100 Punkte abgezogen. Treffen Sie jedoch einen Fisch, so kehrt der Torpedo zu Ihrem U-Boot zurück und schießt dieses ab. Dadurch explodiert das U-Boot und Sie bekommen einen Ihrer erzielten Treffer abgezogen.
Wenn Sie einen Torpedo abgefeuert haben, so können Sie das U-Boot nicht bewegen, solange der Torpedo auf dem Bildschirm sichtbar ist.
Für jedes abgeschossene Schiff bekommen Sie, je nach dessen Geschwindigkeit zwischen 10 und 200 Punkte.

Steuerung:
Das U-Boot bewegt sich mit:
"," - nach links
"." - nach rechts
Zum abfeuern eines Torpedos drücken Sie die Space-Taste.

Der strukturierte Aufbau des Programmes ist aus der beigelegten Zuordnungstafel zu entnehmen.
Hier sind alle Variablen, Basic-Funktionen, Schlüsselwörter, Unterprogramme, Zeilenverweise usw. aufgeführt.

```
4 DATA 1,2,3,4,5
5 CALL TITEL
10 CALL CLEAR :: CALL SCREEN(2):: CALL MAGNIFY(3)
11 GOTO 15 :: B :: I :: IST :: J :: K :: MOT :: P :: PA" :: RICHT :: S :: SOLL :
: T :: TORP :: U :: X :: Y :: CALL ALOHAOE :: CALL CHAR :: CALL COINC :: CALL CO
LOR :: CALL DELSPRITE
12 CALL ENDE :: CALL HCHAR :: CALL KEY :: CALL MOTION :: CALL PATTERN :: CALL PO
SITION :: CALL PUNKTE :: CALL SCHIFF :: CALL SOUND :: CALL SPRITE :: CALL TORPED
O :: !$P-
15 P=0 :: SOLL-4
20 CALL CHAR(88,"0000000101030307070F0F01FFF3331FC0E0F0B0B0B0B0C0E0F0FBB0FFCFCCF
B")
21 CALL CHAR(112,"0003070F3F1-1F3F1F1F0FCF0E060301E0F0D8FEFE700080C0C0B0203C30F0
E0")
22 CALL CHAR(80,"070F1B7F7F0E000103030104000C0F0700C0E0F0F0FBFBFCFBFBF0F07060CCB
0")
23 CALL CHAR(128,"00000001030F17FFFF1F07C3010000000000C0B0B1C2F6FCFCF6C201")
24 CALL CHAR(120,"00000301B1436F3F3F6F43B000000000000000B0C0F0E8FFFFF8F0C0B")
30 CALL CHAR(92,"03070F0101010103070F1F01FFF3331F000000B0B0C0C0E0E0F0F0B0FFCFCCF
B")
35 CALL CHAR(76,"183C3C3C3C181866"):: CALL COLOR(6,14,13)
40 CALL CHAR(96,"0000000000000000101030C37FFFFF7F0F000000000000D0B0B0C0C0FEFFFFFEF
0")
45 CALL CHAR(84,"0000000006010103030303030100000000000000060B0B0C0C0C0C0C0CB")
50 CALL CHAR(100,"0000000000000000101030303030301010600000000000000B0C0C0C0C0C0B0B0
60")
60 CALL CHAR(64,"0000000000000000000000000000A0A0300000000000000000000000000005060C
0")
```

```
70 CALL CHAR(68,"000000000000000040E0E653D1D0E07000000000000000207070A6BCB870E
0")
80 CALL CHAR(72,"80E0783C0E07030383F37F3F1FCF476101071E3C70E0C0C0C1CFFEFCF8F3E28
6")
110 CALL CHAR(104,"000003070F7FF3FF063EFFFFCFBFFDF9000006BFFBFBE7FF0000183CFCFE3
FFC")
120 FOR I=127 TO 143 STEP 8 :: CALL CHAR(I,"FFFFFFFF"):: NEXT I
130 CALL CHAR(119,RPT$("F",16))
140 CALL HCHAR(1,1,127,32):: CALL HCHAR(2,1,135,32):: CALL HCHAR(3,1,143,32)
150 FOR I=0 TO 28 STEP 4 :: FOR J=1 TO 4 :: CALL HCHAR(21,I+J,103+J):: NEXT J ::
 NEXT I
160 FOR I=4 TO 20 :: CALL HCHAR(I,1,119,32):: NEXT I
170 CALL CHAR(111,RPT$("F",16)):: CALL HCHAR(22,1,111,96)
180 CALL COLOR(12,16,12,13,11,10,14,9,7,11,5,1,10,13,5):: CALL SCREEN(13)
185 CALL COLOR(3,16,13,4,16,13)
186 TORP=21 :: CALL PUNKTE(P):: CALL TORPEDO(TORP):: IST=0 :: CALL SCHIFF(SOLL,I
ST)
187 RANDOMIZE :: CALL SPRITE(#6,112,11,48,1,0,INT(RND*10)+6):: RANDOMIZE :: CALL
 SPRITE(#7,80,11,120,240,0,-(INT(RND*10)+6))
188 RANDOMIZE :: CALL SPRITE(#8,128,7,72,240,0,-(INT(RND*10)+6)):: RANDOMIZE ::
CALL SPRITE(#9,120,7,96,1,0,INT(RND*10)+6)
189 CALL SPRITE(#3,96,11,145,120)
190 CALL ALOHADE
191 RANDOMIZE :: RICHT=RND*20
192 MOT=INT(RND*20)+1
193 IF RICHT<10 THEN CALL SPRITE(#1,88,2,9,240,0,-MOT)ELSE CALL SPRITE(#1,92,2,9
,1,0,MOT)
200 CALL PATTERN(#3,96):: CALL COLOR(#3,11):: U=1
201 CALL KEY(5,T,S):: CALL MOTION(#3,0,0):: IF TORP=0 THEN 6000
202 IF S=0 THEN 201
210 IF T=32 THEN 1000
220 IF T=44 THEN 2000
230 IF T=46 THEN 3000 ELSE 200
1000 CALL POSITION(#3,Y,X):: CALL SPRITE(#4,100,14,Y,X):: CALL MOTION(#4,-20,0)
: CALL TORPEDO(TORP)
1100 CALL SOUND(-400,-5,24-Y/8):: CALL POSITION(#4,Y,X)
1200 IF Y<15 THEN CALL DELSPRITE(#4):: GOTO 200
1250 IF Y<132 AND Y>32 THEN CALL COINC(ALL,B):: IF B=-1 THEN 7000
1300 CALL COINC(#1,#4,12,B):: IF B=-1 THEN 5000 ELSE 1100
2000 CALL MOTION(#3,0,-15):: CALL SOUND(-250,110,20,112,20,113,20,-6,15):: GOTO
200
3000 CALL MOTION(#3,0,15):: CALL SOUND(-250,110,20,112,20,113,20,-6,15):: GOTO 2
00
5000 CALL DELSPRITE(#4):: CALL MOTION(#U,0,0)
5005 DATA 64,68,72,68,64
5006 RESTORE 5005
5007 IST=IST+1 :: CALL SCHIFF(SOLL,IST)
5010 FOR I=1 TO 5
5015 READ PAT :: CALL PATTERN(#U,PAT):: CALL COLOR(#U,12)
5020 FOR K=5 TO 0 STEP -1 :: CALL SOUND(-700,-5,K):: NEXT K
5025 CALL COLOR(#U,10)
5030 FOR K=0 TO 5 :: CALL SOUND(-700,-5,K):: NEXT K
5040 NEXT I
5050 CALL DELSPRITE(#1):: P=P+MOT*10 :: CALL PUNKTE(P):: IF TORP=0 THEN 6000 ELS
E 191
6000 IF IST<SOLL THEN CALL ENDE :: GOTO 10
6010 SOLL=SOLL+2 :: IF SOLL>16 THEN SOLL=16
6020 GOTO 186
7000 CALL MOTION(#4,0,0)
7005 FOR I=6 TO 7 :: CALL COINC(#4,#I,20,B):: IF B=-1 THEN 7030
7010 NEXT I
7020 GOTO 8000
7030 CALL DELSPRITE(#4)
7040 CALL COLOR(#I,4)
7050 FOR J=1100 TO 1000 STEP -10 :: CALL SOUND(-250,J,0,J+2,0,J+3,0):: NEXT J
7060 P=P-100 :: IF P<0 THEN P=0
7070 CALL PUNKTE(P):: CALL COLOR(#I,11):: GOTO 200
8000 FOR J=8 TO 9 :: CALL COINC(#4,#J,20,B):: IF B=-1 THEN 8005
8001 NEXT J :: GOTO 9000
8005 CALL PATTERN(#4,84):: CALL MOTION(#4,20,0)
```

```
8010 CALL POSITION(#4,Y,X):: CALL SOUND(-400,-5,24-Y/8)
8020 CALL COINC(#3,#4,12,B):: IF B=-1 THEN 8030 ELSE 8010
8030 MOT=0 :: IST=IST-2 :: IF IST<0 THEN IST=-1
8040 U=3 :: GOTO 5000
9000 CALL DELSPRITE(#4):: GOTO 200
9999 !§P+
10000 SUB PUNKTE(P)
10001 GOTO 10010 :: P :: !§P-
10010 DISPLAY AT(24,23)SIZE(-5):USING "####":P
10015 !§P+
10020 SUBEND
15000 SUB TORPEDO(TORP)
15001 GOTO 15010 :: TORP :: CALL HCHAR :: !§P-
15010 CALL HCHAR(24,1+TORP,32)
15020 TORP=TORP-1 :: IF TORP=0 THEN SUBEXIT
15030 CALL HCHAR(24,2,76,TORP)
15035 !§P+
15040 SUBEND
16000 SUB SCHIFF(SOLL,IST)
16001 GOTO 16010 :: IST :: SOLL :: !§P-
16010 DISPLAY AT(22,20)SIZE(7):USING 16020:IST,SOLL
16020 IMAGE ## (##)
16025 !§P+
16030 SUBEND
20000 SUB EINLEITUNG
20001 GOTO 20010 :: S :: T :: CALL ALOHAOE :: CALL CHAR :: CALL CLEAR :: CALL KE
Y :: CALL PUNKTE :: CALL SCHIFF :: CALL TORPEDO :: !§P-
20010 CALL CLEAR
20020 DISPLAY AT(1,1):"SIE SIND DER KOMANDANT EINESU-BOOTES, UND IHRE AUFGABE  B
ESTEHT DARIN, SO VIELE     SCHIFFE WIE MOEGLICH ABZU-  SCHIESSEN."
20030 DISPLAY AT(6,1):"SCHAFFEN SIE ES NICHT EINE  VORGESEBENE ANZAHL SCHIFFE  (
SOLLVORGABE) IN EINER RUNDEZU VERNICHTEN, SO WERDEN SIE"
20040 DISPLAY AT(10,1):"DEGRADIERT UND DAS SPIEL ISTZU ENDE.
PRO RUNDE STEHEN IHNEN 20   TORPEDOS ZUR VERFUEGUNG."
20045 DISPLAY AT(14,1):"SIE STARTEN EINE NEUE RUNDE,INDEM SIE BEI DER MELODIE
IRGEND EINE TASTE DRUECKEN."
20050 DISPLAY AT(18,1):"STEUERUNG :                    -----------
',' : U-BOOT NACH LINKS   '.'   : U-BOOT NACH RECHTS"
20060 DISPLAY AT(22,1):"SPACE : TORPEDO ABFEUERN" :: DISPLAY AT(24,6):"WEITER MI
T ENTER"
20070 CALL KEY(5,T,S):: IF T=13 THEN 20080 ELSE 20070
20080 CALL CLEAR :: CALL CHAR(76,"183C3C3C3C181866"):: DISPLAY AT(1,1):"PUNKTWER
TUNG :                ---------------"
20090 DISPLAY AT(4,1):"SCHIFF   : 10 - 200 PUNKTE  SEEPFERD : 100 PUNKTE ABZUG F
ISCH    : EIGENES U-BOOT              WIRD ABGESCHOSSEN"
20100 DISPLAY AT(8,1):"            UND SOMIT EIN                TREFFER ABGEZOGEN"
20110 DISPLAY AT(11,1):"ANZEIGE :                        ---------"
20120 CALL PUNKTE(9999):: CALL SCHIFF(12,8):: CALL TORPEDO(21)
20130 DISPLAY AT(16,12):"Punkte----------ö" :: DISPLAY AT(18,12):"Sollvorgabe--ö
  ö" :: DISPLAY AT(17,28):"ö"
20140 DISPLAY AT(19,25):"ö  ö" :: DISPLAY AT(20,12):"Treffer--ö   ö  ö" :: DISPL
AY AT(21,21):"ö   ö  ö"
20150 DISPLAY AT(22,28):"ö" :: DISPLAY AT(23,28):"ö" :: DISPLAY AT(24,27):"-ö" :
: DISPLAY AT(22,1)SIZE(19):"Restliche torpedosö" :: DISPLAY AT(23,19)SIZE(1):"ö"
20160 CALL ALOHAOE
20170 CALL CLEAR
20175 !§P+
20180 SUBEND
29999 SUB ALOHAOE
30000 DATA 392,30000,30000,523,30000,30000,659,196,30000,659,262,330,659,196,300
00,587,262,330
30001 GOTO 30005 :: A :: B :: C :: I :: S :: T :: CALL KEY :: CALL SOUND :: !§P-
30005 DATA 523,220,30000,494,262,349,523,220,30000,440,262,349
30010 DATA 392,196,30000,392,262,330,392,196,30000,392,262,330,392,196,30000,392
,262,330
30015 DATA 659,196,30000,659,262,330,587,196,30000,587,247,349,587,196,30000,554
,247,349
30020 DATA 587,196,30000,587,247,349,698,196,30000,659,247,349,587,196,30000,587
,247,349,587,196,30000,587,247,349
```

```
30025 DATA 587,196,30000,587,247,349
30030 DATA 392,30000,30000,523,30000,30000,659,196,30000,659,262,330,659,196,300
00,587,262,330
30035 DATA 523,220,30000,494,262,349,523,220,30000,440,262,349
30040 DATA 392,196,30000,392,262,330,392,196,30000,392,262,330,392,196,30000,392
,262,330
30045 DATA 523,30000,30000,494,30000,30000,440,220,30000,440,262,349,587,30000,3
0000,523,30000,30000,494,196,30000
30050 DATA 494,247,349,659,30000,30000,587,30000,30000,523,196,30000,523,262,330
30055 DATA 523,196,30000,523,262,330,523,196,30000,523,262,330,523,196,30000
30060 DATA 392,30000,30000,440,220,30000,440,262,349,523,220,30000,523,262,349
30062 DATA 698,220,30000,698,262,349,698,220,30000,440,262,349
30064 DATA 392,196,30000,392,262,330,523,196,30000,523,262,330
30066 DATA 659,196,30000,659,262,330,659,196,30000
30068 DATA 523,262,330,494,196,30000,440,247,349,494,196,30000,523,247,349
30070 DATA 587,196,30000,587,247,349,698,196,30000,698,247,349
30072 DATA 659,196,30000,659,262,330,659,196,30000,659,262,330,523,196,30000,523
,262,330,523,196,30000
30074 DATA 392,30000,30000,440,220,30000,440,262,349,523,220,30000,523,262,349
30076 DATA 698,220,30000,698,262,349,698,220,30000,440,262,349
30078 DATA 392,196,30000,392,262,330,523,196,30000,523,262,330
30080 DATA 659,196,30000,659,262,330,659,196,30000
30082 DATA 523,262,330,494,196,30000,440,247,349,494,196,30000,523,247,349,587,1
96,30000,659,247,349
30084 DATA 587,196,30000,494,247,349,523,196,30000,523,262,330
30086 DATA 523,196,30000,523,262,330,523,196,30000,523,262,330
30100 RESTORE 30000
30200 FOR I=1 TO 128 :: READ A,B,C :: CALL SOUND(300,A,5,B,10,C,10):: CALL KEY(5
,T,S):: IF S<>0 THEN SUBEXIT
30201 NEXT I
30202 GOTO 30100
30203 !$P+
30204 SUBEND
31000 SUB TITEL
31010 DATA 7,112,U,7,129,-,25,95,B,25,112,O,25,129,O,25,146,T
31011 GOTO 31020 :: Z$ :: A :: B :: I :: S :: T :: CALL ALOHADE :: CALL CHAR ::
CALL CLEAR :: CALL COLOR :: CALL DELSPRITE :: CALL EINLEITUNG :: CALL HCHAR :: C
ALL KEY
31012 CALL MAGNIFY :: CALL SCREEN :: CALL SPRITE :: !$P-
31020 CALL CLEAR :: CALL SCREEN(10):: CALL MAGNIFY(2):: CALL COLOR(13,5,5):: CAL
L HCHAR(1,1,128,192):: CALL CHAR(64,"3C4299A1A199423C")
31025 RESTORE 31010
31030 FOR I=23 TO 28 :: READ A,B,Z$ :: CALL SPRITE(#I,ASC(Z$),4,A,B):: NEXT I
31032 DISPLAY AT(10,1):"      WOLLEN SIE EINE          SPIELANLEITUNG (J/N)"
31034 DISPLAY AT(23,1):"  © 1983 BY VOLKER BECKER"
31040 CALL ALOHADE
31045 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL COLOR(13,2,2)
31047 CALL KEY(5,T,S):: IF T=74 THEN CALL EINLEITUNG
31049 !$P+
31050 SUBEND
32000 SUB ENDE
32010 CALL DELSPRITE(ALL):: CALL CHARSET :: CALL SCREEN(2):: CALL CHAR(64,"3C429
9A1A199423C")
32015 CALL COLOR(12,16,12,13,11,10,14,9,7,11,5,1,10,13,5):: CALL SCREEN(13)
32020 DATA 28,112,U,28,129,-,46,95,B,46,112,O,46,129,O,46,146,T
32021 GOTO 32030 :: Z$ :: A :: B :: I :: CALL ALOHADE :: CALL COLOR :: CALL MAGN
IFY :: CALL SPRITE :: !$P-
32030 RESTORE 32020 :: CALL MAGNIFY(2)
32040 FOR I=23 TO 28 :: READ A,B,Z$ :: CALL SPRITE(#I,ASC(Z$),4,A,B):: NEXT I
32050 DISPLAY AT(22,1):"© 1983 BY VOLKER BECKER"
32051 DATA 78,95,G,78,112,A,78,129,M,78,146,E,94,95,O,94,112,V,94,129,E,94,146,R
32052 FOR I=1 TO 8 :: READ A,B,Z$ :: CALL SPRITE(#I,ASC(Z$),16,A,B):: NEXT I
32053 DISPLAY AT(16,1):" FUER EIN NEUES SPIEL BITTE  IRGEND EINE TASTE DRUECKEN
"
32059 CALL ALOHADE
32060 CALL DELSPRITE(ALL):: CALL CHARSET :: CALL SCREEN(2)
32065 !$P+
32070 SUBEND
```

ZUORDNUNGS-TAFEL DES PROGRAMMS U-Boot

PROGRAMMTEIL (HAUPTPROGRAMM)

NUMERISCHE_VARIABLE

```
B          11   1250  1300  7085  8000  8020
I          11   120   150   160   5010  5040  7085  7010
           7040 7070
IST        11   186   5007  6000  8030
J          11   150   7050  8000  8001
K          11   5020  5030
MIT        11   192   193   5050  8030
P          11   15    186   5050  7060  7070
PAT        11   5015
NICHT      11   191   193
S          11   201   202
SOLL       11   15    186   5007  6000  6010
T          11   201   210   220   230
TORP       11   186   201   1000  5050
U          11   200   5000  5015  5025  6040
X          11   1000  1100  8010
Y          11   1000  1100  1200  1250  6010
```

1

BASIC-SCHLUESSELWORTE

```
A (DATEI:  187   188   189   193   200   201   1000  1100
           1200  1300  2000  3000  5000  5015  5025  5050
           7000  7085  7030  7040  7070  8000  8005  8010
           8020  9000
           16    11    12    15    35    120   140   150
           160   170   180   186   187   188   191   200
           201   1000  1100  1200  1250  1300  2000  3000
           5000  5007  5015  5020  5025  5050  6000  6010
           7005  7050  7060  7070  8000  8001  8005  8010
           8020  8030  8040  9000
ALL        1250
CALL       5     10    11    12    20    21    22    23
           24    30    35    40    45    50    60    70
           80    110   120   140   150   160   170
           180   185   186   187   188   189   190   193
           200   201   1000  1100  1200  1250  1300  2000
           3000  5000  5007  5015  5020  5025  5030  5050
           6000  7000  7005  7030  7040  7050  7070  8000
           8005  8010  8020  9000
DATA       4     5005
ELSE       193   230   1300  5050  8020
FOR        120   150   160   5010  5020  5030  7005  7050
           8000
GOTO       11    1200  2000  3000  1000  4020  7020  7070
           8001  8340  9000
IF         193   201   202   210   220   230   1200  1250
           1300  5050  6000  6010  7005  7060  8000  8020
           8030
NEXT       120   150   160   5020  5030  5040  7010  7050
           8001
RANDOMIZE  187   188   191
READ       5015
RESTORE    5006
STEP       120   150   5020  7050
```

BASIC-FUNKTIONEN

```
*          187   188   191   192   5030
+          150   187   188   192   5007  5050  6010  7050
           187   188   193   1000  1100  1250  1300  2000
           3000  5020  5030  7005  7050  7060  8000  8010
           8020  8030
/          1100  6010
<          193   1200  1250  6000  7060  8030
=          15    120   150   160   186   191   192   200
           201   202   210   220   230   1250  1300  5007
           5010  5020  5030  5050  6010  7005  7050  7060
           8000  8020  8030  8040
>          1250  6010
AND        1250
INT        187   188   192
RND        187   188   191   192
RPT$       190   170
```

UNTERPROGRAMME

```
A.CHARS    11    190
CHAR       11    20    21    22    23    24    30    35
           40    45    50    60    70    80    110   120
           130   170
CLEAR      10
COINC      11    1250  1300  7005  8000  8020
COLOR      11    35    180   185   200   5015  5025  7010
           7070
DILSPRITE  11    1200  5000  5050  7030  9000
ENDE       12    6000
HCHAR      12    140   150   160   170
KEY        12    201
MAGNIFY    10
MOTION     12    201   1000  2000  3000  5000  7000  8005
PATTERN    12    200   5015  8005
POSITION   12    1000  1100  8010
PUNKTE     12    186   5050  7070
SCHIFF     12    186   5007
SCREEN     10    160
SOUND      12    1100  2000  3000  5020  5030  7050  8010
SPRITE     12    187   188   189   193   1000
TITEL      3
TORPEDO    12    186   1000
```

ZEILENREFERENZEN

```
10         6000

1000       210

1100       1300
15         11
186        8020
191        5050

200        230   1200  2000  3000  7070  9000

2000       220
201        201

3000       230

5000       1300  6040
5005       5006

6000       201   5050

7000       1250
7030       7005

8000       7020
8005       8000
8010       8020
8030       8020

9000       8001
```

2

PROGRAMMTEIL PUNKTE

NUMERISCHE_VARIABLE

```
P          10000 10001 10010
```

BASIC-SCHLUESSELWORTE

```
:          10001
AT         10010
DISPLAY    10010
GOTO       10001
SIZE       10010
SUB        10000
SUBEND     10020
USING      10010
```

BASIC-FUNKTIONEN

```
           10010
```

ZEILENREFERENZEN

```
10010      10001
```

3

PROGRAMMTEIL TORPEDO

NUMERISCHE_VARIABLE

```
TORP       15000 15001 15010 15020 15030
```

BASIC-SCHLUESSELWORTE

```
:          15001 15020
CALL       15001 15010 15030
GOTO       15001
IF         15020
SUB        15000
SUBEND     15040
SUBEXIT    15020
```

BASIC-FUNKTIONEN

```
+          15010
-          15020
=          15020
```

UNTERPROGRAMME

```
HCHAR      15001 15010 15030
```

ZEILENREFERENZEN

```
15010      15001
```

PROGRAMMTEIL SCHIFF

NUMERISCHE VARIABLE

IST	16000 16001 15010
SOL	16000 16001 15010

BASIC-SCHLUESSELWORTE

..	16001
AT	16010
DISPLAY	16010
GOTO	16001
IMAGE	16020
SIZE	16010
SUB	16000
SUBEND	16050
USING	16010

ZEILEN-REFERENZEN

16010	16001
16020	16010

4

PROGRAMMTEIL EINLEITUNG

NUMERISCHE VARIABLE

S	20001 20070
T	20001 20070

BASIC-SCHLUESSELWORTE

..	20001 20060 20070 20080 20120 20130 20140 20150
AT	20020 20030 20040 20045 20050 20060 20080 20090 20100 20110 20130 20140 20150
CALL	20001 20030 20070 20080 20120 20160 20170
DISPLAY	20020 20030 20040 20045 20050 20060 20080 20090 20100 20110 20130 20140 20150
ELSE	20070
GOTO	20001
IF	20070
SIZE	20150
SUB	20080
SUBEND	20160

BASIC-EUROPAEISCH

-	20070

UNTERPROGRAMME

ALPHABET	20001 20160
CHAR	20001 20060
CLEAR	20001 20010 20080 20170
KEY	20001 20070
PUNKTE	20001 20120
SCHIFF	20001 20120
TORPEDO	20001 20120

ZEILEN-REFERENZEN

20010	20001
20070	20070
20050	20070

ZUORDNUNGS-TAFEL DES PROGRAMMS U 8084 01.12.83 U-200178

PROGRAMMTEIL ALPHABET

NUMERISCHE VARIABLE

A	30001 30200
B	30001 30200
C	30001 30200
I	30001 30200 30201
S	30001 30200
T	30001 30200

BASIC-SCHLUESSELWORTE

..	30001 30200
CALL	30001 30200
DATA	30000 30005 30010 30015 30020 30025 30030 30035 30040 30045 30050 30055 30060 30062 30064 30066 30068 30070 30072 30074 30076 30078 30080 30082 30084 30086
FOR	30200
GOTO	30001 30202
IF	30200
NEXT	30201
READ	30200
RESTORE	30100
SUB	23999
SUBEND	30284
SUBEXIT	30200

5

BASIC-EUROPAEISCH

<	30200
-	30200
>	30200

ZEILEN-REFERENZEN

30000	30100
30005	30001
30100	30202

UNTERPROGRAMME

KEY	30001 30200
SOUND	30001 30200

PROGRAMMTEIL TITEL

SIEHE-VARIABLE

Z#	31011 31030

NUMERISCHE VARIABLE

A	31011 31030
B	31011 31030
I	31011 31030
S	31011 31047
T	31011 31047

BASIC-SCHLUESSELWORTE

M (DATEI)	31030
..	31011 31012 31028 31030 31045 31047
ALL	31045
AT	31032 31034
CALL	31011 31012 31023 31030 31040 31045 31047
DATA	31010
DISPLAY	31032 31034
FOR	31030
GOTO	31011
IF	31047
NEXT	31030
READ	31030
RESTORE	31025
SUB	31000
SUBEND	31050

BASIC-EUROPAEISCH

*	31030 31047
ASC	31030

6

UNTERPROGRAMME

ALPHABET	31011 31040
CHAR	31011 31020
CLEAR	31011 31020 31045
COLOR	31011 31020 31045
DELSPRITE	31011 31045
EINLEITUNG	31011 31047
MONAT	31011 31020
KEY	31011 31047
MAGNIFY	31012 31020
SCREEN	31012 31020
SPRITE	31012 31030

ZEILEN-REFERENZEN

31010	31020
31020	31011

PROGRAMMTEIL ENDE

SIEHE-VARIABLE

Z#	32021 32040 32052

NUMERISCHE VARIABLE

A	32021 32040 32052
B	32021 32040 32052
I	32021 32040 32052

BASIC-SCHLUESSELWORTE

M (DATEI)	32040 32052
..	32010 32015 32021 32030 32040 32052 32060
ALL	32010 32060
AT	32050 32053
CALL	32010 32015 32021 32030 32040 32052 32059 32060
DATA	32020 32051
DISPLAY	32050 32053
FOR	32040 32052
GOTO	32021
NEXT	32040 32052
READ	32040 32052
RESTORE	32030
SUB	32000
SUBEND	32070

BASIC-EUROPAEISCH

-	32040 32052
ASC	32040 32052

7

UNTERPROGRAMME

ALPHABET	32021 32060
CHAR	32010
CHARSET	32010 32060
COLOR	32015 32021
DELSPRITE	32010 32060
MAGNIFY	32021 32030
SCREEN	32010 32015 32060
SPRITE	32021 32040 32052

ZEILEN-REFERENZEN

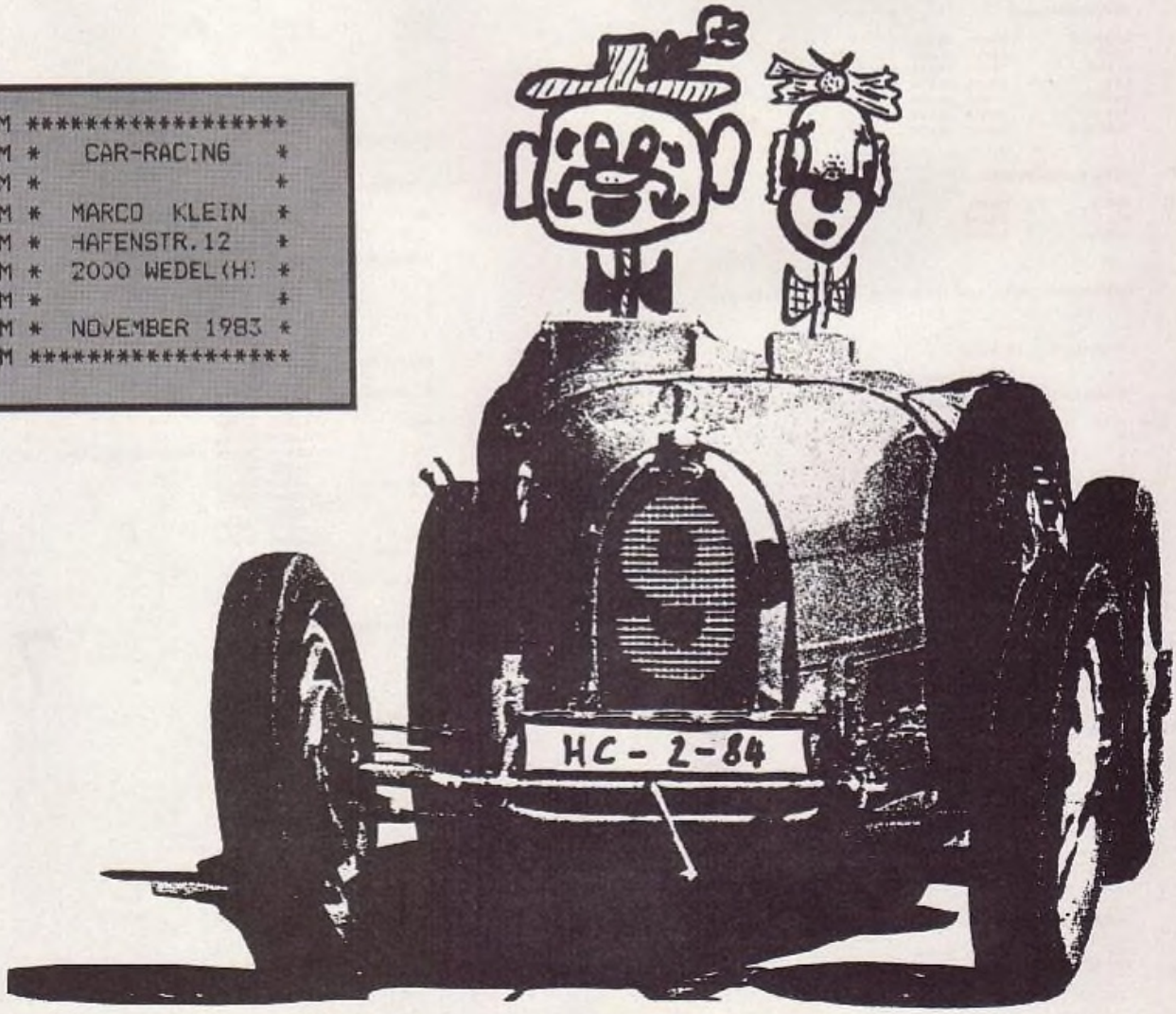
32010	32030
32030	31021

Car-Racing

für den TI-99/4A

Da wir bisher in Homecomputer noch kein Autorennen für den TI-99/4A veröffentlicht haben, möchten wir dies mit dem Programm "Car Racing" schleunigst nachholen.
Da Autorennspiele immer noch mit die beste Möglichkeit bieten, sein Reaktionsvermögen zu testen, werden bestimmt auch Sie von unserem Spiel begeistert sein, aber vielleicht ganz am Anfang auch einwenig entnervt, zumindest bis Sie sich eingefahren haben.
Ein Programm mit sehr guter Graphik, das über Joystick gespielt wird.
Als Fahrer eines Rennwagens, muß man versuchen, den ständig auf der Strecke auftauchenden Hindernissen und den entgegenkommenden Autos auszuweichen.
Was anfangs wahrscheinlich als viel zu leicht empfunden wird, steigert sich hinterher zu einer Horrorfahrt ins Ungewisse.

```
10 REM ******************
20 REM *   CAR-RACING   *
30 REM *                *
40 REM *  MARCO  KLEIN  *
50 REM *  HAFENSTR.12   *
60 REM *  2000 WEDEL(H) *
70 REM *                *
80 REM *  NOVEMBER 1983 *
90 REM ******************
```



```
100 P=0 :: A=3 :: S=0 :: R=1
110 CALL CHAR(96,"7EFFC37E7EFFE77E")
120 CALL COLOR(9,5,1)
125 CALL CLEAR :: FOR I=2 TO 8 :: CALL COLOR(I,16,1):: NEXT I :: CALL SCREEN(2)
130 DISPLAY AT(7,6):"C A R - R A C I N G" :: DISPLAY AT(8,6):'==================
="
140 CALL SOUND(1000,220,0):: CALL SOUND(200,200,0):: FOR I=1 TO 6 :: CALL SOUND(
100,330,0):: NEXT I
150 CALL SOUND(600,265,0):: FOR I=1 TO 4 :: CALL SOUND(100,330,0):: NEXT I :: CA
LL SOUND(1000,220,0)
160 CALL CLEAR :: CALL SCREEN(2)
170 CALL MAGNIFY(3)
180 CALL CHAR(140,"1F3F3FFFFFF0F0303F3F3FFFF0F0FF7FF8FCFCFFFF0F0F0CFCFCFCFF0F0FF
FFE")
190 CALL VCHAR(1,16,122,24*12):: CALL COLOR(12,11,11)
200 CALL VCHAR(1,20,119,24*4):: CALL VCHAR(1,12,119,24*4):: CALL COLOR(11,13,13)
210 CALL CHAR(104,"141F6B566DE35DA6AD567F3E7F2D1C0BC8B44CAA6EAA55AE5FABD5BAF5EED
0B8")
220 CALL SPRITE(#2,104,4,1,12*8):: CALL SPRITE(#3,104,4,80,12*8):: CALL SPRITE(#
4,104,4,160,12*8)
230 CALL SPRITE(#5,104,4,1,28*8):: CALL SPRITE(#6,104,4,80,28*8):: CALL SPRITE(#
7,104,4,160,28*8)
240 CALL SPRITE(#1,140,5,21*8,20*8)
250 DISPLAY AT(9,2)SIZE(8):"PUNKTE:"
260 DISPLAY AT(10,1)SIZE(8):"";P
270 DISPLAY AT(16,2)SIZE(8):"KM/H:"
280 DISPLAY AT(17,1)SIZE(8):"";G
290 DISPLAY AT(2,2)SIZE(8):"STRECKE:"
300 DISPLAY AT(3,1)SIZE(8):"";S
310 CALL HCHAR(21,4,96):: DISPLAY AT(21,3)SIZE(3):"";A
320 CALL SPRITE(#8,140,7,1,17*8)
330 CALL SPRITE(#9,140,2,1,23*8)
340 CALL SPRITE(#10,140,3,1,160)
350 CALL JOYST(1,X,Y):: CALL MOTION(#1,0,4*X):: CALL COINC(ALL,C):: IF C THEN 46
0
360 CALL MOTION(#9,5+G,0):: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,4*X):: CALL COI
NC(ALL,C):: IF C THEN 460
370 CALL MOTION(#8,2+G,0):: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,4*X):: CALL COI
NC(ALL,C):: IF C THEN 460
380 CALL MOTION(#10,4+G,0):: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,4*X):: CALL CO
INC(ALL,C):: IF C THEN 460
390 FOR I=2 TO 7 :: CALL MOTION(#I,G,0):: NEXT I :: CALL JOYST(1,X,Y):: CALL MOT
ION(#1,0,4*X):: CALL COINC(ALL,C):: IF C THEN 460
400 G=G+.5*R :: IF G>120 THEN G=120 :: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,4*X)
:: CALL COINC(ALL,C):: IF C THEN 460
410 CALL POSITION(#1,C,D):: IF D<15*8 OR D>25*8 THEN 460
420 DISPLAY AT(17,1)SIZE(8):"";4*G :: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,4*X):
: CALL COINC(ALL,C):: IF C THEN 460
430 S=S+2*G :: DISPLAY AT(3,1)SIZE(5):"";S :: IF S/R>=2000 THEN 520
440 P=P+10 :: DISPLAY AT(10,1)SIZE(7):"";P :: CALL JOYST(1,X,Y):: CALL MOTION(#1
,0,4*X):: CALL COINC(ALL,C):: IF C THEN 460
450 GOTO 350
460 CALL COLOR(#1,7):: CALL PATTERN(#1,104):: CALL MOTION(#1,0,0):: CALL SOUND(6
00,-5,0):: FOR I=1 TO 150 :: NEXT I
470 G=0 :: A=A-1 :: DISPLAY AT(21,3)SIZE(3):"";A :: IF A=0 THEN 500 :: CALL PATT
ERN(#1,140):: CALL COLOR(#1,5):: CALL LOCATE(#1,21*8,20*8)
480 FOR I=1 TO 300 :: CALL KEY(0,K,ST):: IF ST THEN 350
485 NEXT I
490 GOTO 350
500 CALL DELSPRITE(#1):: DISPLAY AT(12,15)SIZE(9):"GAME OVER" :: CALL KEY(0,K,S)
:: IF S=0 THEN 500 ELSE 510
510 CALL DELSPRITE(ALL):: FOR I=1 TO 400 :: NEXT I :: GOTO 100
520 CALL DELSPRITE(ALL):: CALL SOUND(1000,220,0):: FOR I=1 TO 4 :: CALL SOUND(30
0,330,0):: NEXT I :: CALL SOUND(1000,220,0)
530 G=0 :: R=R+1 :: A=A+1 :: GOTO 160
```

TI ärgere Dich nicht für den TI-99/4A

Nach dem Programmstart wird zunächst eine Kurzfassung der Spielanleitung auf dem Bildschirm angezeigt. Anschließend können der/die Spieler entscheiden, ob ein neues Spiel begonnen oder ein altes Spiel zu Ende gespielt werden soll. Im letzteren Fall müssen die vorher auf Band gespeicherten Daten vom Kassetten-Recorder übertragen werden.

Außer dem bekannten Spielfeld mit blauen, roten, violetten und grünen Figuren, werden weitere für die Spielsteuerung wichtige Informationen dargestellt:

1. Auf der linken Seite die Zeilen-Nummern (Y) von 0-10
2. Unter dem Spielfeld die Spalten-Nummern (X) von 0-10
3. Den Spaltennummern 0-4 sind Farbsymbole und der Buchstabe "W" zugeordnet.

Beim Spielstart blinken bei gleichzeitigem Piepen die Ziffern 0-3. Der Spieler wird damit aufgefordert, die der Farbe seiner Wahl entsprechende Taste zu drücken (z.B. 1 für Rot). Über der zugehörigen Warteposition erscheint eine Zahl zwischen 1-6.

Folgende Spielsituationen können jetzt vorliegen:

a) Der Spieler hat keine "6" gewürfelt und keine Figur im Spiel. Das Programm springt wieder auf Farbenwahl zurück.

b) Es wurde eine "6" gewürfelt und keine Figur im Spiel. Unter dem Spielfeld werden die Zeilennummer "Y" und die Spaltennummer "X" des Standortes der Figur abgefragt, die man setzen will. Nach jeder Eingabe "Enter" drücken. Die gewählte Figur wird in einen doppelt großen, gelben Sprite umgewandelt. Anschließend müssen die "Zielkoordinaten" des Feldes eingegeben werden, das der gewürfelten Zahl entspricht (die Spielregeln dieses Spieles werden als bekannt vorausgesetzt). Bei einer "6" darf der Spieler nochmals würfeln.

Falsche Eingaben, z.B. Startpositionen von fremden Farben oder unbesetzten Feldern, werden kurzfristig mit "Falsche Koordinaten" angezeigt. Entsprechendes wird bei den Zielkordinaten angezeigt, wenn die Position außerhalb des Spielfeldes liegt, oder die Zielfelder fremder Farben in der Spielmitte angewählt werden.

Geschlagene Figuren wandern zu ihren Wartepositionen zurück. Hat ein Spieler alle Figuren auf den Zielfeldern untergebracht, wird der Bildschirm gelöscht und die Siegerfarbe angezeigt, z. B. "Violett hat gewonnen". Mit (9) kann man ein neues Spiel starten.

Besonderheiten:

Falsche Spielzüge können wie folgt mit "Shift" (=) korrigiert werden:

1. Falsches Setzen, ohne eine Figur zu schlagen: Sofort nach dem Würfeln, aber vor der Wahl einer neuen Farbe, die o. a. Tasten drücken. Das Programm fragt direkt nach der Ausgangsposition. Es werden die falschen Koordinaten eingeben und anschließend die Daten des richtigen Feldes.
2. Es wurde durch falsches Setzen eine Figur geschlagen: Nach der oben beschriebenen Korrektur wählt man die entsprechende Farbe und rückt die Tasten "Shift"(=) vor dem Würfeln. Die geschlagene Figur kann wieder eingesetzt werden. Der interne zugehörige Zähler wird wieder um 1 erhöht.
3. Je nach Auslegung der Spielregeln darf man, wenn ein Teil der Figuren einer Farbe die Spielfelder erreicht hat und der Rest der gleichen Farbe auf den Wartepositionen steht, nicht mehr setzen. Das Programm berücksichtigt diese Regel, indem nach dem Würfeln einer Zahl unter "6" eine neue Farbe gewählt werden muß. Mit "Shift"(=) kann man diese Regelung umgehen, um Figuren im Ziel bei entsprechend kleiner "Augenzahl" zusammenrücken zu können. Die Tasten nach dem Würfeln, aber vor einer neuen Farbenwahl drücken.
4. Speichern des Spielstandes:

Bei vorzeitigem Spielabbruch kann man den Spielstand vor einer Farbenwahl mit (?) auf Band speichern. Nach Drücken der Taste wird der Bildschirm abgetastet. Anschließend erfolgen die Befehle für die Steuerung des Recorders. Man sollte die Daten direkt im Anschluß an das Programm speichern. Damit erübrigt sich der "Rewind-Befehl" beim Speichern bzw. bei der Entscheidung für die Fortführung eines alten Spieles nach Laden des Programmes.

5. Es ist nur eine Figur einer Farbe auf dem Spielfeld und steht kurz vor dem Ziel. Wenn die gewürfelte Zahl größer ist als die Reststrecke, kann nicht gesetzt werden. Bei der ersten Koordinaten-Abfrage, wird dann bei "Y" ein Wert kleiner "0" eingegeben. Das Spiel fährt mit Farbenwahl fort, wenn die Zahl kleiner "6" war.

Variablen-Liste

BL1, RO1, LT1, GR1 = Anzahl der zur Verfügung stehenden Figuren

BL, RO, LI, GR = Anzahl der freien Zielfelder

BLAU, ROT, LILA, GRUEN = Anzahl der sich im Spiel befindlichen Figuren

B,R,L,G = Symbole der Farben bei der Bestimmung der freien Zielfelder

Die Variablen werden in den Programmzeilen 495 - 2005 berechnet. Das Programm entscheidet hier, ob eine "6" geworfen wurde und erhöht dann beim Einsetzen einer Figur den entsprechenden Zähler, z. B. BLAU = BLAU + 1. Mit BL, RO, LI, GR kann die Anzahl der zur Verfügung stehenden Figuren neu festgelegt werden. Um die Anzahl der besetzten Zielfelder werden BL1, RO1, LI1 oder GR1 reduziert. Wenn BLAU, ROT, LILA oder GRUEN = 0, kann nicht gesetzt werden und der nächste Spieler ist an der Reihe.

Z1 - Z5 = Nicht erlaubte Zielfelder und Farben.

Mit diesen Variablen werden bei den Ausgangskoordinaten alle fremden Farben und bei dem Zielkoordinaten alle fremden Zielfelder gesperrt. Das heißt, wurde "ROT" gewählt, kann nicht mit "Blau", "Violett" oder "Gruen" gespielt oder deren End-Position angewählt werden.

Q = Würfelwurf 1-6

FI = Figur auf dem Spielfeld "A"

Verhindert bei mehrmaligem Würfeln von "6" hintereinander eine Erhöhung des Zählers (z.B. ROT), wenn eine eigene Figur auf dem Feld "A" steht.

PY, PX = Koordinaten für die Markierung der gewählten Farbe.

X1, Y1 = Über die Tastatur eingegebenen Spalten - und Zeilennummern für die Ausgangsposition einer Figur.

X2, Y2 wie bei X1, Y1, nur für Zielkoordinaten.

X3, X4, Y3, Y4, Positionsmeldungen des Sprites bei der Wanderung von der Ausgangsposition zum Ziel.

F = Symbol für das Startfeld.

Nach dem Abzug einer Figur wird das bisher verdeckte Feld wieder sichtbar, z. B. F = 65 für das Feld "A".

T = Symbol des mit X1, Y1 angewählten Zeichens.

T1 = Symbol des mit X2, Y2 angewählten Zeichens.

T1 entscheidet, ob das Programm z. B. auf Sieg abgefragt oder eine Fehlermeldung (z. B. bei falscher Farbe) anzeigt.

D8 (Z) = Bildschirmzeile bei der Bildschirmabfrage für die Speicherung eines aktuellen Spielstandes bei vorzeitigem Spielabbruch.

SCHALTER = Entscheidung, ob Spielfeld neu aufgebaut oder altes Spiel übernommen wird.

```
1 REM `
2 REM VON HORST BUSSE
3 REM A.D.STALLBERG 27 D
4 REM 5 KOELN 90
10 CALL CLEAR
20 PRINT "ES HANDELT SICH UM EIN ":"BEKANNTES BRETTSPIEL,":"DESSEN REGELN BEKANN
T SEIN":"DUERFTEN."
21 PRINT "DAS SPIEL WIRD UEBER DIE":"TASTATUR GESTEUERT.MIT DEN  TASTEN:" ::
30 PRINT "          <0> <1> <2> <3>":"WIRD DIE FARBE GEWAEHLT." "DIE TASTE FESTHAL
TEN BIS":"EIN SIGNAL ERTOENT":"GEWUERFELT WIRD MIT <4>'
40 PRINT "DIE TASTE FESTHALTEN BIS EIN":"SIGNAL ERTOENT.":"BEI EINER 6 KANN DER
SPIELER":"EINE FIGUR SEINER FARBE"
41 PRINT "DURCH EINGABE VON KOORDINA-":"TEN  AUSWAEHLEN.":"DIE AUGENZAHL WIRD IN
 DER   SPIELFELDMITTE ANGEZEIGT!"
50 PRINT :: PRINT "WEITER MIT <W>"
51 CALL KEY(0,K,S):: IF S=0 THEN 51 :: IF K<>87 THEN 51 :: CALL CLEAR
61 PRINT "ZUNAECHST WIRD DIE ZEILEN-":"NUMMER(Y) UND DANN DIE ":"SPALTENNUMMER(X
) DER":"AUSGANGSPOSITION EINGEGEBEN."
62 PRINT "GENAUSO WIRD MIT DEN ZIEL-":"KOORDINATEN VERFAHREN.":"DIE GEWAEHLTE FI
GUR WANDERT UEBER DAS SPIELFELD ZUM GE-":"WUENSCHTEN ZIEL."
63 PRINT "POSITIONEN FREMDER FAREEN":"ODER KOORDINATEN AUSSERHALB":"DES SPIELFEL
DES WERDEN ALS":"FEHLER KURZFRISTIG UNTER DEM"
64 PRINT "SPIELFELD ANGEZEIGT.ES":"MUESSEN NEUE DATEN EIN-":"GEGEBEN WERDEN."
65 PRINT "WEITER MIT <W>"
66 CALL KEY(0,K,S):: IF S=0 THEN 66 :: IF K<>87 THEN 66 :: CALL CLEAR
67 PRINT "DER JEWEILIGE SPIELSTAND":"KANN VOR DER FARBENWAHL MIT":"<T> GESPEICHE
RT WERDEN."
68 PRINT "FALSCHE KOORDINATEN KOENNEN DURCH <+> VOR DER NEUEN FAR-BENWAHL KOORIG
IERT WERDEN.":"FALSCH GESCHLAGENE FIGUREN"
69 PRINT "WERDEN NACH DER FARBENWAHL":"ABER VOR DEM WUERFELN" "MIT <+> EINGESETZ
T":
70 PRINT :: PRINT :: PRINT :"DRUECKE <N> FUER NEUES SPIEL":"DRUECKE <D> FUER ALT
ES SPIEL"
75 PRINT :: PRINT 'REWIND BEFEHL NICHT BEFOLGEN':'DA DIE DATEN AM PROGRAMMENDE':
"GESPEICHERT WERDEN"
80 DIM D$(24)
90 CALL KEY(0,K,S):: IF S=0 THEN 90 :: IF K=68 THEN 9640 :: IF K=78 THEN 99 ELSE
 90
99 BL1,RO1,LI1,GR1=4  :: BLAU,ROT,LILA,GRUE=0 :: SCHALTER=0
100 DATA 3C3C1813183C7EFF,3C7EFFFFFFFF7E3C
130 REM DEFINITION DER ZEICHEN UND FARBEN
140 CALL CLEAR :: CALL SCREEN(2):: CALL CHAR(33,"FFFFFFFFFFFFFFFF",40,"000000FFF
F",41,'1818181813181818',36,"181818C3C3181818")
145 CALL MAGNIFY(2)
150 READ A$ :: FOR I=0 TO 4 :: CALL CHAR(96+I,A$,104+I,A$,112+I,A$,120+I,A$):: N
EXT I
160 READ B$ :: FOR I=0 TO 24 STEP 8 :: CALL CHAR(101+I,B$):: NEXT I
180 CALL COLOR(0,16,1,1,16,1,2,14,1,9,5,1,10,7,1,11,14,1,12,13,1,13,2,15) : FOR
I=3 TO 8 :: CALL COLOR(I,2,15):: NEXT I
185 IF SCHALTER=1 THEN 9690
190 REM AUFBAU DES SPIELFELDES
200 DATA " 0  \ a      !<!<A      h i","              )   )","  l  b c      | m |
J k","              )   )"
210 DATA " 2            | m |","              )   )"," 3            | m |","
  )   )"
220 DATA " 4  A<!<!<!<! m !<!<!<!<!","              )                  )"," 5  | e e e e
  u u u u |"
230 DATA "              )                  )"," 6  !<!<!<!<! ) !<!<!<!<A","              )
  )
240 DATA " 7            | ) |","              )   )"," 8            | ) |","
  )   )"," 9  x y      | ) |      p q","              )   )"
250 DATA "10  z (      A<!<!      r s"
260 FOR I=1 TO 21 :: READ F$ :: DISPLAY AT(I,1):F$ :: NEXT I
270 C$="d l t | W "
280 DISPLAY AT(23,5):"0 1 2 3 4 5 6 7 8 9 10"
290 CALL DELSPRITE(#2):: DISPLAY AT(24,1):"  "&C$
300 REM SPIEL
310 CALL KEY(0,K,ST):: IF ST=0 THEN DISPLAY AT(23,5)SIZE(7)BEEP :: CALL SOUND(10
00,20000,30):: DISPLAY AT(23,5)SIZE(7):"0 : 2 3" :: GOTO 310
311 IF K=63 THEN 9500
314 IF K=43 THEN 3000
315 IF K<48 OR K>51 THEN 310
317 CALL SOUND(300,700,0,300,0)
320 ON K-47 GOTO 495,995,1495,1995
495 X=9 :: BL=0
496 CALL GCHAR(11,X,B):: IF B=101 THEN BL=BL+1
497 X=X+2 :: IF X<17 THEN 496
498 IF BL<BL1 THEN BLAU=BLAU-1 ELSE 500
499 IF BL<BL1 THEN BL1=BL1-1
500 Z1=96 :: Z2=101 :: Z3=109  : Z4=117 :: Z5=125 :: PY=6 :: PX=54 :: GOSUB 2500
 :: IF Q<6 AND BLAU=0 THEN 290 :: IF Q<6 AND BLAU>0 THEN 3000
503 IF Q=6 AND BLAU<BL THEN 506 ELSE 3000
506 CALL GCHAR(9,7,FI):: IF FI=96 OR FI=97 OR FI=98 OR FI=99 THEN 3000 ELSE BLAU
```

```
=BLAU+1 :: GOTO 3000
995 Y=3 :: RO=0
996 CALL GCHAR(Y,17,R):: IF R=109 THEN RO=RO+1
997 Y=Y+2 :: IF Y<11 THEN 996
998 IF RO<RO1 THEN ROT=ROT-1 ELSE 1000
999 IF RO<RO1 THEN RO1=RO1-1
1000 Z1=104 :: Z2=109 :: Z3=101 :: Z4=117 :: Z5=125 :: PY=6 :: PX=198 :: GOSUB 2
500 :: IF Q<6 AND ROT=0 THEN 290 :: IF Q<6 AND ROT>0 THEN 3000
1003 IF Q=6 AND ROT<RO THEN 1005 ELSE 3000
1005 CALL GCHAR(1,19,FI):: IF FI=104 OR FI=105 OR FI=106 OR FI=107 THEN 3000 ELS
E ROT=ROT+1 :: GOTO 3000
1495 X=19 :: LI=0
1496 CALL GCHAR(11,X,L):: IF L=117 THEN LI=LI+1
1497 X=X+2 :: IF X<27 THEN 1496
1498 IF LI<LI1 THEN LILA=LILA-1 ELSE 1500
1499 IF LI<LI1 THEN LI1=LI1-1
1500 Z1=112 :: Z2=117 :: Z3=101 :: Z4=109 :: Z5=125 :: PY=150 :: PX=198 :: GOSUB
 2500 :: IF Q<6 AND LILA=0 THEN 290 :: IF Q<6 AND LILA>0 THEN 3000
1503 IF Q=6 AND LILA<LI THEN 1505 ELSE 3000
1505 CALL GCHAR(13,27,FI):: IF FI=112 OR FI=113 OR FI=114 OR FI=115 THEN 3000 EL
SE LILA=LILA+1 :: GOTO 3000
1995 Y=13 :: GR=0
1996 CALL GCHAR(Y,17,G):: IF G=125 THEN GR=GR+1
1997 Y=Y+2 :: IF Y<21 THEN 1995
1998 IF GR<GR1 THEN GRUE=GRUE-1 ELSE 2000
1999 IF GR<GR1 THEN GR1=GR1-1
2000 Z1=120 :: Z2=125 :: Z3=101 :: Z4=109 :: Z5=117 :: PY=150 :: PX=54 :: GOSUB
2500 :: IF Q<6 AND GRUE=0 THEN 290 :: IF Q<6 AND GRUE>0 THEN 3000
2003 IF Q=6 AND GRUE<GR THEN 2005 ELSE 3000
2005 CALL GCHAR(21,15,FI):: IF FI=120 OR FI=121 OR FI=122 OR FI=123 THEN 3000 EL
SE GRUE=GRUE+1 :: GOTO 3000
2500 CALL SPRITE(#2,36,16,PY,PX):: CALL KEY(0,KEY,ST):: IF ST=0 THEN DISPLAY AT(
23,13)SIZE(1)BEEP :: CALL SOUND(1000,20000,30)
2501 DISPLAY AT(23,13)SIZE(1):"4"
2505 IF KEY=43 THEN 2600
2510 IF KEY<>52 THEN 2500 :: CALL WUR(Q):: RETURN
2600 IF K=48 THEN BLAU=BLAU+1 :: GOTO 3000
2610 IF K=49 THEN ROT=ROT+1 :: GOTO 3000
2620 IF K=50 THEN LILA=LILA+1 :: GOTO 3000
2630 IF K=51 THEN GRUE=GRUE+1
3000 ON ERROR 8020 :  ON WARNING NEXT
3005 CALL SPRITE(#1,36,11,193,1):: DISPLAY AT(24,1):"KOORDINATEN:" :: DISPLAY AT
(24,18):"Y:" :: ACCEPT AT(24,20)VALIDATE(DIGIT,"-")BEEP:Y1
3007 IF Y1<0 THEN 3081 :: DISPLAY AT(24,23):"X:"
3009 ON ERROR 8020 :: ON WARNING NEXT
3010 ACCEPT AT(24,25)VALIDATE(DIGIT)BEEP:X1 :: CALL GCHAR((Y1+1)*2-1,X1*2+7,T)::
 IF T<Z1 OR T>=Z2 THEN 8000
3011 CALL PATTERN(#1,T):: CALL LOCATE(#1,Y1*16+1,X1*16+45):: F=33 :: GOSUB 4400
3019 ON ERROR 8030 :: ON WARNING NEXT
3020 CALL HCHAR((Y1+1)*2-1,X1*2+7,F):: DISPLAY AT(24,1):"ZIELKOORDINATEN:" :: DI
SPLAY AT(24,19):"Y:" :: ACCEPT AT(24,21)VALIDATE(DIGIT)BEEP:Y2
3030 DISPLAY AT(24,24):"X:" :: ACCEPT AT(24,26)VALIDATE(DIGIT)BEEP:X2
3045 CALL GCHAR((Y2+1)*2-1,X2*2+7,T1):: IF T1=Z3 XOR T1=Z4 XOR T1=Z5 THEN 8010 :
: IF T1<96 THEN 3050 ELSE GOSUB 4500
3046 GOTO 3070
3050 IF T1=65 THEN GOSUB 4500 ELSE 3051 :: GOTO 3070
3051 IF T1=33 THEN GOSUB 4500 ELSE 8010
3070 CALL HCHAR((Y2+1)*2-1,X2*2+7,T):: IF T1=33 OR T1=65 OR T1=Z2 THEN 3080 ELSE
 4000
3080 CALL LOCATE(#1,193,126):: IF T1=Z2 THEN 9000
3081 IF Q<6 THEN 290
3090 DISPLAY AT(24,1):"              W" :: GOTO 320
4000 Y1=Y2 :: X1=X2 :: CALL PATTERN(#1,T1)!AUSGANGSPOSITION FUER RUECKFUEHRUNG D
ER FIGUR
4005 ON T1-95 GOTO 4010,4020,4030,4040,1,4170,1,1,4050,4060,4070,4080,1,4170,1,1
,4050,4100,4110,4120,1,4170,1,1,4130,4140,4150,4160,1,4170
4010 Y2=0 :: X2=0 :: BLAU=BLAU-1 :: GOSUB 4500 :: GOTO 4165 !BLAUE FIGUREN
4020 Y2=0 :: X2=1 :: BLAU=BLAU-1 :: GOSUB 4500 :: GOTO 4165
4030 Y2=1 :: X2=0 :: BLAU=BLAU-1 :: GOSUB 4500 :: GOTO 4165
4040 Y2=1 :: X2=1 :: BLAU=BLAU-1 :: GOSUB 4500 :: GOTO 4165
4050 Y2=0 :: X2=9 :: ROT=ROT-1 :: GOSUB 4500 :: GOTO 4165 !ROTE FIGUREN
4060 Y2=0 :: X2=10 :  ROT=ROT-1 :: GOSUB 4500 :: GOTO 4165
4070 Y2=1 :: X2=9 :: ROT=ROT-1 :: GOSUB 4500 :: GOTO 4165
4080 Y2=1 :: X2=10 :  ROT=ROT-1 :: GOSUB 4500 :: GOTO 4165
4090 Y2=9 :: X2=9 :: LILA=LILA-1 :: GOSUB 4500 :: GOTO 4165 !LILA FIGUREN
4100 Y2=9 :: X2=10 :: LILA=LILA-1 :: GOSUB 4500 :: GOTO 4165
4110 Y2=10 :: X2=9 :: LILA=LILA-1 :: GOSUB 4500 :: GOTO 4165
4120 Y2=10 :: X2=10 :: LILA=LILA-1 :: GOSUB 4500 :: GOTO 4165
4130 Y2=9 :: X2=0 :: GRUE=GRUE-1 :: GOSUB 4500 :: GOTO 4165 !GRUENE FIGUREN
4140 Y2=9 :: X2=1 :: GRUE=GRUE-1 :: GOSUB 4500 :: GOTO 4165
4150 Y2=10 :: X2=0 :: GRUE=GRUE-1 :: GOSUB 4500 :: GOTO 4165
4160 Y2=10 :: X2=1 :: GRUE=GRUE-1 :: GOSUB 4500
4165 CALL HCHAR((Y2+1)*2-1,X2*2+7,T1):: CALL SOUND(1000,560,0,345,0):: CALL SOUN
```

```
D(450,8000,0)
4170 GOTO 3080
4400 REM ZEICHEN DES ALTEN FELDES
4410 IF Y1=4 AND X1=0 THEN F=65
4411 IF Y1=0 AND X1=6 THEN F=65
4412 IF Y1=6 AND X1=10 THEN F=65
4413 IF Y1=10 AND X1=4 THEN F=65
4414 IF Y1=5 AND X1<5 THEN F=101
4415 IF Y1=5 AND X1=0 THEN F=33
4416 IF Y1=5 AND X1>5 THEN F=117
4417 IF Y1=5 AND X1=10 THEN F=33
4418 IF X1=5 AND Y1<5 THEN F=109
4419 IF X1=5 AND Y1>5 THEN F=125
4420 IF X1=5 AND Y1=10 THEN F=33
4421 IF Y1=0 AND X1<2 THEN F=101
4422 IF Y1=1 AND X1<2 THEN F=101
4423 IF Y1=0 AND X1>8 THEN F=109
4424 IF Y1=1 AND X1>8 THEN F=109
4425 IF Y1=9 AND X1<2 THEN F=125
4426 IF Y1=10 AND X1<2 THEN F=125
4427 IF Y1=9 AND X1>8 THEN F=117
4428 IF Y1=10 AND X1>8 THEN F=117
4430 IF X1=5 AND Y1=0 THEN F=33
4431 RETURN
4500 X3=X2*16+45 :: Y3=Y2*16+1 :: X4=X1*16+45 :: Y4=Y1*16+1
4510 IF X4<X3 THEN X4=X4+8
4520 IF Y4<Y3 THEN Y4=Y4+8
4530 IF X4>X3 THEN X4=X4-8
4540 IF Y4>Y3 THEN Y4=Y4-8
4550 CALL LOCATE(#1,Y4,X4):: IF Y4=Y3 AND X4=X3 THEN RETURN ELSE 4510
7999 REM FEHLERMELDUNG
8000 CALL SOUND(600,4000,0,8000,0,500,0):: DISPLAY AT(24,1)BEEP:"FALSCHE KOORDIN
ATEN!" :: FOR I=1 TO 200 :: NEXT I :: GOTO 3000
8010 CALL SOUND(600,8000,0,1200,0):: DISPLAY AT(24,1):"FALSCHE KOORDINATEN!" ::
FOR I=0 TO 200 :: NEXT I :: GOTO 3020
8020 DISPLAY AT(24,1):"Y UND/ODER X ZU GROSS!" :: CALL SOUND(500,8000,0,1200,0):
: FOR I=1 TO 200 :: NEXT I :: GOTO 3000
8030 DISPLAY AT(24,1):"Y UND/ODER X ZU GROSS!" :: CALL SOUND(500,8000,0,1200,0):
: FOR I=1 TO 200 :: NEXT I :: GOTO 3019
9000 REM SIEG
9005 ZF=0
9010 IF T1=101 THEN M,X=9 ELSE 9020 :: GOTO 9050
9020 IF T1<>109 THEN 9030 :: IF T1=109 THEN M,Y=3 :: GOTO 9070
9030 IF T1=117 THEN M,X=19 ELSE 9040 :: GOTO 9050
9040 IF T1=125 THEN M,Y=13 :: GOTO 9070
9050 CALL GCHAR(11,X,Z6):: IF Z6=T1 THEN ZF=ZF+1
9060 X=X+2 :: IF X<M+8 THEN 9050 ELSE 9090
9070 CALL GCHAR(Y,17,Z7):: IF Z7=T1 THEN ZF=ZF+1
9080 Y=Y+2 :: IF Y<M+8 THEN 9070
9090 IF ZF=0 THEN 9095 ELSE 3081
9095 CALL DELSPRITE(#2)
9100 IF T1=101 THEN DISPLAY AT(12,5)ERASE ALL:"BLAU HAT GEWONNEN" ELSE 9110 :: G
OTO 9200
9110 IF T1=109 THEN DISPLAY AT(12,5)ERASE ALL:"ROT HAT GEWONNEN" ELSE 9120 :: GO
TO 9200
9120 IF T1=117 THEN DISPLAY AT(12,5)ERASE ALL:"LILA HAT GEWONNEN" ELSE 9130 :: G
OTO 9200
9130 IF T1=125 THEN DISPLAY AT(12,5)ERASE ALL:"GRUEN HAT GEWONNEN"
9200 CALL SCREEN(12):: FOR I=200 TO 1000 STEP 50 :: CALL SOUND(200,I,0,3000-I,0)
:: NEXT I
9201 DISPLAY AT(23,4):"NEUES SPIEL? JA(9)  NEIN(0)"
9210 CALL KEY(0,K,S):: IF S=0 THEN 9210
9220 IF K=48 THEN 10090 :: IF K=57 THEN 9230 ELSE 9210
9230 RESTORE :: GOTO 99
9500 REM SPEICHERN DES SPIELSTANDES
9520 FOR Z=1 TO 21 :: FOR SPALTE=3 TO 30 :: CALL GCHAR(Z,SPALTE,C):: D$(Z)=D$(Z)
&CHR$(C):: NEXT SPALTE :: NEXT Z
9580 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED 192
9590 FOR I=1 TO 20 STEP 5 :: PRINT #1:D$(I),D$(I+1),D$(I+2),D$(I+3),D$(I+4):: NE
XT I
9620 PRINT #1:D$(21),BLAU,ROT,LILA,GRUE,BL1,RO1,LI1,GR1
9630 CLOSE #1
9635 CALL CLEAR :: GOTO 9201
9640 OPEN #1:"CS1",INTERNAL,INPUT ,FIXED 192
9650 FOR I=1 TO 20 STEP 5 :: INPUT #1:D$(I),D$(I+1),D$(I+2),D$(I+3),D$(I+4):: NE
XT I
9680 INPUT #1:D$(21),BLAU,ROT,LILA,GRUE,BL1,RO1,LI1,GR1
9682 CLOSE #1
9683 RESTORE :: SCHALTER=1 :: GOTO 100
9690 FOR I=1 TO 21 :: DISPLAY AT(I,1):D$(I):: NEXT I
9720 GOTO 270
10000 SUB WUR(Q):: CALL SOUND(1000,400,0,500,6):: RANDOMIZE :: Q=INT(6*RND)+1 ::
 DISPLAY AT(11,14)SIZE(2):Q :: SUBEND
10090 END
```

TI 99/4A

Antares
für den TI-99/4A

Als Teilnehmer eines Forschungstrupps haben Sie den Auftrag, den Höhlenplaneten Antares zu erkunden und eine Bodenprobe desselben zu entnehmen.

Die Teleportationskugel materialisiert sich kurz über der Oberfläche von Antares und besitzt 3 Energieschilde, die bei Kollision jeweils zerstört werden.
Hat man erfolgreich den Abstieg in den Planeten überstanden, kommt man in eine großräumige Höhle, welche zu durchqueren ist. Dieses Vorhaben wollen jedoch zahlreiche Lasergeschütze verhindern. Schafft man es trotzdem, gelangt man nun nach einem weiteren, längerem Abstieg in eine Tropfsteingrotte, die ebenfalls durchquert werden muß, unglücklicherweise ist hier aber der Schlafplatz der gefürchteten Riesenfledermäuse von Antares, die natürlich nun sehr aufgeregt umherflattern. Ist dieses Abenteuer auch überstanden, wird es Zeit, eine Bodenprobe von Antares zu nehmen (auf der Plattform rechts unten).
Da Sie sich jetzt im Mittelpunkt des Planeten befinden, gelten nun die Gesetze der Schwerelosigkeit, also Vorsicht bei zu hoher Beschleunigung.
Zusätzlich macht Ihnen ein antaresischer Kampfflieger das Leben schwer. Um ihm zu entgehen, gibt es nur eine bestimmte Taktik, die wir aber nicht verraten wollen. Ist man dann endlich, mit der niedrigsten Geschwindigkeit die möglich ist, glücklich gelandet (zwecks Bodenprobe), taucht ein ganzes Geschwader von Kampffliegern auf, denen Sie ausweichen müssen.
Nachdem all dies bewältig ist, hat man zum guten Schluß, nach einer Verschnaufpause, noch einen Abstieg zu bewältigen, um endlich auf der anderen Seite des Planeten herauszukommen.

Anmerkung: Kleinbuchstaben müssen als Kleinbuchstaben eingegeben werden.
ß steht für @, ´ steht für #, § steht für $

```
1ØØ CALL CLEAR :: CALL SCREE
N(2) :: DIM B§(4Ø),V(2Ø),H(2Ø
),C(2Ø),F(5):: BEGINN=Ø :: T
OP=Ø
11Ø GOTO 43Ø
12Ø B§(1)="ßßßßßßßßßßßßßßß
ßA  ßßßßßßß" :: B§(2)="ßßßßß
GßßßGGßßßßßßGF  EGGßßßß" ::
B§(3)="ßßßGF EGF   EGGGF
    E3ß"
13Ø B§(4)="ßßF
          ßß" :: B§(5)="ßA
                        ßß" ::
B§(6)="ßF
    Eß"
14Ø B§(7)="A
          ß" :: B§(8)="ßI
                        ß" ::
B§(9)="ßßD
    Cß"
15Ø B§(1Ø)="ßßßI
          ßß" :: B§(11)="ßßß
A                       ßß" :
: B§(12)="ßßßßI
          ß"
16Ø B§(13)="ßßßßF
          Cß" :: B§(14)="ßßß
ßI                        ßß" :
: B§(15)="ßßßßA
          ßß"
17Ø B§(16)="ßßßßßI
          ßß" :: B§(17)="ßßß
ßßF                       Cßß" :
: b§(18)="ßßßFI
          ßßß"
18Ø B§(19)="ßßßßA
          Cßßßß" :: B§(2Ø)="ßßß
ßßI                       Cßßßßßß" :
: B§(21)="ßßßßßßD
  ßßßßßßßß"
19Ø B§(22)="ßßßßßFI          Cß
ßßßßßßßßßßßß" :: B§(23)="ßßß
ßßßßD      Cßßßßßßßßßßßßß3" :
: B§(24)="ßßßßßßßßD   Cßßßßß
ßßßß3ßßßßß"
2ØØ RETURN
21Ø B§(1)="ßßA  ßßßßßßßßßßß
ßßßßßßßßßßß" :: B§(2)="ßßF
EGßßßßßßßßßßßßßßßßßßß3ßß" ::
B§(3)="ßF      EßßGF  EßßßßßA
EGßßßßß"
22Ø B§(4)="A          EF      ßß
ßßßF  EGGßß" :: B§(5)="ßD
          Eßßßß      Eß"
23Ø B§(6)="ßßGDCßD   C      ß
ßßF        ß" :: B§(7)="ßßßßß
ßA   A     ßßA          ß" ::
B§(8)="ßßßGFßA   ßD    Eßß
    Cß"
24Ø B§(9)="ßßA   A  CßA
ßA        ßß" :: B§(1Ø)="ßßßI
  F  ßßA     ßA          Eß" ::
B§(11)="ßßßA    Cßßßß    EF
        ß"
25Ø B§(12)="ßßßF     ßßßßA
 ß        Cßßß" :: B§(13)="ßßA
    Cßßßßß    F    Cßßßß" :
: B§(14)="ßßßA  ßßßßßßßA
  ßßßßßFßß"
26Ø B§(15)="EßßA CßßßßßßßD
   CGF EA Eß" :: B§(16)="ßßß
ßßßßßßßßßßßßßD  F    ACßß" :
: B§(17)=" EßßßßßßßEßßßßßßF
    Eßßß"
27Ø B§(18)="C3ßßßßßßßA ßßßßßA
          ßßß" :: B§(19)="ßßß
ßßßßßGF  ßßßßßßßD       Eßß" :
: B§(2Ø)="ßßßßßßA      Eßßßßßß
A         ß"
28Ø B§(21)="ßßßßßßßßßD   Eßß
ßßßßD      Cß" :: B§(22)="ßßß
ßßßßßßßßßD  Cßßßßß   Cßßß" :
: B§(23)="ßßßßßßßßßßßßßßßßß
ßßA   ßßßß"
29Ø RETURN
3ØØ FOR I=2Ø TO 25 :: B§(I)=
"" :: NEXT I
31Ø B§(1)="ßßßßßßßßßßßßA  ßß
ßßßßßßßßßßß" :: B§(2)="ßßßßß
ßßßßßGGF  EGGßßßßßßßßßß" ::
B§(3)="ßßßßßßßGGF          EGG
ßßßßßßß"
32Ø B§(4)="ßßßGGGF
   EGGGßßß" :: B§(5)="ßßF
                        Eßß" ::
B§(6)="ßF
    E3"
33Ø B§(7)="A
        B" :: B§(8)=B§(7):
: B§(9)="F
        E"
```

```
34Ø B§(3Ø)="DMCD MOCBBDNNOCB
BBDMM LL CBB" :: B§(31)=RPT§
("B",28):: B§(32)=B§(31)
35Ø REM
36Ø GOTO 2Ø2Ø
37Ø REM
38Ø B§(1)="
          C" :: B§(2)="
                       B" ::
B§(3)="
     CB"
39Ø B§(4)="D
        CBB" :: B§(5)="A
                      BBB" ::
B§(6)="BBD     D
   CBBB"
4ØØ B§(7)="BBBD  CA      CD
        BBBB" :: B§(8)="BBBA
CBBD     BBD        CCBBBB" ::
B§(9)="BBBBBBBBBD  CBBBD
CBBBBBB"
41Ø B§(1Ø)="BBBBBBBBBBBBBBBB
A     BBBBBBB"
42Ø RETURN
43Ø DEF YP=((Y-1)/8)+1 :: DE
F XP=((X-1)/8)+1
44Ø RANDOMIZE :: Q=1 :: LAE(
1)=6Ø :: LAE(2)=4Ø :: LAE(3)
=1Ø
45Ø V(1)=1Ø :: H(1)=7 :: C(1
)=22 :: F(1)=5 :: F(2)=6 ::
F(3)=8 :: F(4)=15 :: FA(1)=7
 :: FA(2)=14 :: FA(3)=9
46Ø V(6)=12 :: H(6)=8 :: C(6
)=22
47Ø V(7)=14 :: H(7)=8 :: C(7
)=21
48Ø V(5)=18 :: H(5)=8 :: C(5
)=2Ø
49Ø V(4)=2Ø :: H(4)=9 :: C(4
)=15
5ØØ V(3)=22 :: H(3)=1Ø :: C(
3)=7
51Ø V(2)=16 :: H(2)=9 :: C(2
)=2Ø
52Ø V(8)=8 :: H(8)=5 :: C(8)
=25
53Ø REM
54Ø CALL CHAR(64,"FFFFFFFFFF
FFFFFFFCF8FEFCFFF8FCF81F3F1F
FF3F7F1F3F")
55Ø CALL CHAR(67,"ØØØ1Ø5ØFØF
3F7FFF",68,"ØØCØEØE8F8FCFEFF
FF3F1F17Ø7Ø3Ø1Ø1FFFEFAFØFØCØ
8Ø8ØFFFFFFFFFFFFFF65")
56Ø CALL CHAR(72,"DB7E3C1818
18ØØØØØØØØ7EFEBØFCAEFF181818
181818181818ØØØØFFFF")
57Ø CALL CHAR(76,"ØØØØØØØØØØ
ØØFFFFØØ386CFE6C385492ØØØØØØ
ØØØ3337FFFØØØØØØF8C8C9FFFFF")
58Ø CALL CHAR(128,"3C66FF3DB
DFF663C")
59Ø CALL CHAR(129,"ØØØØØ8Ø24
9Ø327ØF",13Ø,"ØØ1Ø4ØØØ94CØE8
FØ",131,"FØE4CØ924Ø1ØØØØØ",1
33,"ØF17Ø329ØØØ2Ø8")
6ØØ C§="CØCØ6Ø6734FFFC78" ::
 D§="ØØØ7Ø7FEFC783Ø6Ø" :: CA
LL CHAR(132,D§)
61Ø CALL CHAR(134,"1EØ41CFF1
CØ41EØØØØØØØØØØØFFFFØØ")
62Ø FOR I=96 TO 12Ø STEP 8 :
: CALL CHAR(I,"FFFFFFFFFFFFF
FFF"):: NEXT I
63Ø CALL COLOR(5,13,1,6,11,1
,13,11,1,8,16,1)
64Ø CALL COLOR(1,8,1,2,13,1,
3,7,1,4,7,1,9,5,1,1Ø,11,1,11
,7,1,12,14,1)
65Ø CALL CHAR(92,"ØØØØ1",93,
"ØØØØØØØØØØØØ8",94,"ØØØØØØØ81
CØ8")
66Ø L=16 :: GOSUB 37Ø
67Ø IF BEGINN=1 THEN 7ØØ
68Ø PRINT "```  `  ` ```  ```
 ```  ```  ```  ```  ```  ```
 B B B   B hhh hhhh h   hhh
hhh hh hhhp p p pp p   p p
pp p   p"
69Ø PRINT "x x x x x   x x
x x xxx xxx" :: PRINT ::PRI
NT :: PRINT
7ØØ FOR I=1 TO 1Ø :: PRINT
B§(I):: B§(I)="" :: NEXT I
71Ø IF BEGINN=1 THEN 79Ø
72Ø CALL CHAR(113,"ØØØØØØ784
478Ø4Ø4Ø")
73Ø DISPLAY AT(12,9)SIZE(12)
:"von m.qelgen"
74Ø FOR I=1 TO 3Ø
75Ø XS=INT(RND*26)+3 :: YS=I
NT(RND*22)+1
76Ø CALL GCHAR(YS,XS,Z):: IF
 Z<>32 THEN 75Ø
77Ø CALL HCHAR(YS,XS,INT(RND
*3)+92):: NEXT I
78Ø GOSUB 275Ø :: GOTO 66Ø
79Ø CALL SPRITE(#5,128,2,17,
121)
8ØØ CALL SOUND(2ØØ,22ØØ,6,2Ø
ØØ,6,18ØØ,6)
81Ø FOR I=1 TO 3Ø :: CALL CO
LOR(#5,16):: CALL COLOR(#5,7
):: NEXT I
82Ø CALL COLOR(#5,F(Q)):: Y=
121
83Ø X=17 :: FOR I=1 TO LAE(1
):: K=INT(RND*3)-1
84Ø L=L+K
85Ø IF L<3 THEN L=L+1 ELSE I
F L>22 THEN L=L-1
86Ø A§=RPT§("B",L)&"A   B"&
RPT§("B",22-L)
87Ø PRINT A§
88Ø GOSUB 296Ø
89Ø NEXT I
9ØØ X=17
91Ø FOR I=1 TO LAE(2):: K=IN
T(RND*3)+1
92Ø L=L+K
93Ø IF L<3 THEN L=L+1 ELSE I
F L>22 THEN L=L-1
94Ø A§=RPT§("B",L)&"A   B"&R
PT§("B",23-L)
95Ø PRINT A§
96Ø GOSUB 296Ø :: NEXT I
97Ø REM
98Ø FOR I=1 TO LAE(3):: K=IN
T(RND*3)-1
99Ø L=L+K
1ØØØ IF L<3 THEN L=L+1 ELSE
IF L>22 THEN L=L-1
1Ø1Ø A§=RPT§("B",L)&"A B"&R
PT§("B",24-L)
1Ø2Ø PRINT A§
1Ø3Ø GOSUB 295Ø :: NEXT I
1Ø4Ø IF RU=1 THEN 155Ø ELSE
IF RU=2 THEN 188Ø ELSE IF RU
=3 THEN 274Ø
1Ø5Ø GOSUB 12Ø :: IF L<18 TH
EN 11ØØ
1Ø6Ø FOR I=L TO 18 STEP -1 :
: L=L-1 :: A§=RPT§("B",L)&"A
 B"&RPT§("B",24-L)
```
```

```
1070 PRINT A$ :: GOSUB 2960
1080 NEXT I
1090 GOTO 1150
1100 FOR I=L TO 18
1110 L=L+1
1120 A$=RPT$("B",I)&"A  B"&R
PT$("B",24-L)
1130 PRINT A$
1140 GOSUB 2960 :: NEXT I
1150 REM
1160 FOR I=1 TO 23 :: PRINT
B$(I):: B$(I)=""
1170 GOSUB 2960
1180 NEXT I
1190 IF RU=1 THEN 1630
1200 DISPLAY AT(24,1):B$(24)
1210 REM
1220 X=17
1230 CALL LOCATE(#5,X,Y):: I
F XP=24 THEN 1440
1240 GE=INT(RND*8)+1
1250 CALL HCHAR(V(GE),H(GE),75
75,C(GE))
1260 GOSUB 1380
1270 REM
1280 CALL JOYST(2,A,B):: Y=Y
+(2*A):: X=X-(2*B)
1290 CALL GCHAR(XP,YP,Z):: I
F Z<>32 THEN GOSUB 1310
1300 GOTO 1230
1310 REM
1320 Q=Q+1 :: CALL SOUND(300
,-7,0,180,0):: FOR J=1 TO 50
:: NEXT J :: IF Q=5 THEN 13
30 ELSE 1370
1330 F(Q)=16 :: CALL POSITIO
N(#5,X,Y):: CALL DELSPRITE(#
5)
1340 CALL SPRITE(#1,129,F(Q)
,X-4,Y-4,-8,-8,#2,130,F(Q),X
-4,Y+4,-8,8,#3,131,F(Q),X+4,
Y+4,8,8,#4,133,F(Q),X+4,Y-4,
8,-8)
1350 CALL SOUND(300,-6,0,110
,8)
1360 FOR J=1 TO 100 :: NEXT
J :: CALL DELSPRITE(#1,#2,#3
,#4)::FOR I=1 TO 500 :: NEX
T I :: CALL CLEAR :: GOTO 28
80
1370 CALL COLOR(#5,F(Q)):: R
ETURN
1380 IF V(GE)<>XP THEN 1420
1390 CALL COLOR(#5,7)
1400 CALL SOUND(100,110,1)
1410 GOSUB 1310
1420 CALL HCHAR(V(GE),H(GE),
32,C(GE))
1430 CALL COLOR(#5,F(Q)):: R
ETURN
1440 REM
1450 REM
1460 L=8
1470 FOR I=1 TO 21 :: K=INT(
RND*3)-1
1480 L=L+K :: X=X-8
1490 IF L<3 THEN L=L+1 ELSE
IF L>22 THEN L=L-1
1500 A$=RPT$("B",L)&"A  B"&
RPT$("B",23-L)
1510 PRINT A$
1520 CALL LOCATE(#5,X,Y):: N
EXT I
1530 REM
1540 LAE(2)=50 :: LAE(3)=15
:: RU=1 :: GOTO 910
1550 FOR I=L TO 2 STEP -1
1560 K=INT(RND*3)-1
1570 L=I
1580 A$=RPT$("B",L)&"A  B"&R
PT$("B",24-L)
1590 PRINT A$
1600 GOSUB 2960 :: NEXT I
1610 GOSUB 210
1620 GOTO 1160
1630 DISPLAY AT(24,1):"BBBBB
BBBBBBBBBBBBBBBA  BBBB"
1640 REM
1650 FOR I=25 TO 161 STEP 8
1660 GESCH=INT(RND*7)+4
1670 CALL SPRITE(#((I-1)/8)+
3,132,FA(INT(RND*3)+1),I,1,0
,GESCH):: NEXT I
1680 X=17
1690 CALL CHAR(132,D$)
1700 CALL COINC(ALL,BUM):: I
F BUM=-1 THEN GOSUB 1310
1710 GOSUB 2070
1720 REM
1730 CALL CHAR(132,C$):: CAL
L COINC(ALL,BUM):: IF BUM=-1
 THEN GOSUB 1310
1740 GOSUB 2070
1750 GOTO 1690
1760 REM
1770 GOSUB 1310
1780 REM SIEG
1790 CALL DELSPRITE(ALL):: C
ALL SPRITE(#5,128,F(Q),x,y):
: L=20
1800 FOR I=1 TO 21 :: K=INT(
RND*3)-1
1810 L=L+K :: X=X-8
1820 IF L<3 THEN L=L+1 ELSE
IF L>22 THEN L=L-1
1830 A$=RPT$("B",L)&"A  B"&
RPT$("B",23-L)
1840 PRINT A$
1850 CALL LOCATE(#5,X,Y):: N
EXT I
1860 REM
1870 LAE(3)=20 :: RU=2 :: GO
TO 910
1880 IF L<12 THEN 1940
1890 FOR I=L TO 12 STEP -1
1900 L=I
1910 A$=RPT$("B",L)&"A  B"&R
PT$("B",24-L)
1920 PRINT A$
1930 GOSUB 2960 :: NEXT I
1940 REM
1950 FOR I=L TO 11
1960 L=I
1970 A$=RPT$("B",L)&"A  B"&R
PT$("B",24-L)
1980 PRINT A$
1990 GOSUB 2960 :: NEXT I
2000 REM
2010 GOTO 300
2020 FOR I=1 TO 32
2030 PRINT B$(I):: B$(I)=""
2040 GOSUB 2960 :: NEXT I
2050 REM
2060 GOTO 2110
2070 CALL JOYST(2,A,B):: Y=Y
+(2*A):: X=X-(2*B)
2080 CALL GCHAR(XP,YP,Z):: I
F Z<>32 THEN GOSUB 1310
2090 CALL LOCATE(#5,X,Y)
2100 IF XP=24 THEN 1780 :: R
ETURN
2110 REM
2120 REM
2130 CALL SPRITE(#1,76,14,16
1,193,#2,76,14,161,201,#3,13
```

```
4,14,17,254,2,-30)
2140 REM
2150 CALL JOYST(2,A,B):: JO=
(A*10)+B
2160 IF JO<>4 THEN 2180
2170 XG=XG-1 :: GOTO 2240
2180 IF JO<>40 THEN 2200
2190 YG=YG+1 :: GOTO 2240
2200 IF JO<>-4 THEN 2220
2210 XG=XG+1 :: GOTO 2240
2220 IF JO<>-40 THEN 2240
2230 YG=YG-1
2240 CALL MOTION(#5,XG,YG)
2250 CALL COINC(#5,#3,10,BUM
2260 IF BUM=-1 THEN 2330
2270 CALL POSITION(#5,X,Y)::
CALL MOTION(#5,0,0):: IF Y>
160 THEN 2300
2280 CALL MOTION(#5,XG,YG)
2290 GOTO 2150
2300 REM
2310 IF Y<192 OR Y>209 THEN
GOSUB 1310 ELSE IF XG>2 THEN
 GOSUB 1310 ELSE 2350
2320 REM
2330 CALL MOTION(#5,0,0):: G
OSUB 1310 ::
2340 GOTO 2270
2350 REM
2360 CALL DELSPRITE(#1,#2,#3
):: A$=""
2370 F$="DMCD MOCBBDNNOCBBBD
MM LL CBBBBD NCDMMCB3DOOONCB
BBBBDMNO    CDMM N    OO CBBB
D CD  N NCBBBD M          N OC
D M    N            O CBBBD  M"
2380 E$="CD  CBBD M   N CD"
2390 A$=F$&E$ :: CALL POSITI
ON(#5,X,Y)
2400 CALL HCHAR(9,23,135,8):
: FOR I=1 TO 20
2410 DISPLAY AT(21,1)SIZE(28
):SEG$(A$,I,28)
2420 CALL LOCATE(#5,X,Y):: Y
=Y-8
2430 CALL HCHAR(9,23-I,135):
: NEXT I
2440 REM
2450 X=153 :: Y=41
2460 CALL LOCATE(#5,X,Y)
2470 REM
2480 FOR I=73 TO 153 STEP 8
2490 CALL SPRITE(#((I-1)/8)+
1,134,INT(RND*13)+3,I,256,0,-
(-INT(RND*11)+20))
2500 NEXT I
2510 FOR TIME=1 TO 300
2520 CALL JOYST(2,A,B):: X=X
-(2*B)
2530 CALL LOCATE(#5,X,Y)
2540 CALL COINC(ALL,BUM)
2550 IF BUM=-1 THEN GOSUB 13
10
2560 IF X>153 OR X<73 THEN G
OSUB 1310 ELSE 2570
2570 NEXT TIME
2580 REM
2590 CALL CHAR(134,"000000018
18"):: FOR I=10 TO 20 :: CAL
L DELSPRITE(#I):: NEXT I
2600 CALL HCHAR(9,1,32,32)::
 ZEI=20
2610 CALL JOYST(2,A,B):: X=X
-(2*B):: ZEI=ZEI+1 :: Y=Y+(2
*A)
2620 DISPLAY AT(21,1)SIZE(28
):SEG$(A$,ZEI,28):: CALL LOC
ATE(#5,X,Y)
2630 IF ZEI=LEN(A$)-15 THEN
2650
2640 GOTO 2610
2650 REM
2660 M$="BBBBBBBBBBBBBBBBBBB
BBBBBBB3BD    CBB3BB" :: N$="
BBBBBBBBBBBBBBBBBBBBBBBBBBB
A    BBBBBB"
2670 FOR I=1 TO 10
2680 DISPLAY AT(21,1)SIZE(28
):SEG$(A$,ZEI+I,28):: DISPLA
Y AT(22,1)SIZE(28):SEG$(M$,I
,28):: DISPLAY AT(22,1)SIZE(
28):SEG$(N$,I,28)
2690 DISPLAY AT(23,1)SIZE(28
):SEG$(N$,I,28):: NEXT I
2700 A$="" :: M$="" :: N$=""
2710 FOR I=X TO 17 STEP -8 :
: CALL LOCATE(#5,I,Y):: NEXT
I
2720 RU=3 :: LAE(2)=60 :: LA
E(3)=20 :: I=19 :: GOTO 900
2730 REM
2760 FOR TON=1 TO 3 :: FOR I
=110 TO 1100 STEP 110 :: CAL
L SOUND(100,I,0):: NEXT I
2770 FOR I=1100 TO 110 STEP
110 :: CALL SOUND(100,I,0):
: NEXT I
2780 FOR I=110 TO 1100 STEP
110 :: CALL SOUND(100,I,0,12
10-I,0):: NEXT I :: NEXT TON
2790 IF TOP=1 THEN RETURN EL
SE CALL CLEAR
2800 FOR I=1 TO 15 :: PRINT
:: NEXT I :: BEGINN=1 ::RET
URN
2810 FOR I=1 TO 23 :: PRINT
2820 GOSUB 2960
2830 NEXT I
2840 CALL DELSPRITE(ALL)
2850 TOP=1
2860 PRINT "          ''''  '  '
' '''                B   BB B B
 B B                 hhh hhhh h
 h hhh               p   p pp p
 p p                 xxx x  x x
x  xxx"
2870 FOR I=1 TO 10 :: PRINT
:: NEXT I :: GOSUB 2750 :: G
OTO 2930
2880 REM
2890 CALL CLEAR :: CALL DELS
PRITE(ALL)
2900 FOR I=110 TO 1100 STEP
110 :: CALL SOUND(100,I,0,I+
10,0,I+20,0):: NEXT I
2910 FOR I=110 TO 880 STEP 2
0 :: CALL SOUND(-100,I,0,990
-I,0):: NEXT I
2920 DISPLAY BEEP AT(12,1):"
    g a m e   o v e r    "
2930 DISPLAY BEEP AT(22,1):"
neue runde (j/n) ?"
2940 CALL KEY(0,KE,ST):: IF
ST=0 THEN 2940 ELSE IF KE=10
6 THEN 3000 ELSE CALL CLEAR
2950 END
2960 CALL JOYST(2,A,B):: Y=Y
+(2*A)
2970 CALL GCHAR(XP,YP,Z):: I
F Z<>32 THEN GOSUB 1310
2980 CALL LOCATE(#5,X,Y)
2990 RETURN
3000 CALL CLEAR :: BEGINN=1
:: TOP=0 :: GOTO 430
```

Wanderung
für den TI-99/4A

Nach dem Vorspann des Programmes erscheint ein Rasterbild, ein rotes, grünes und ein blaues Zeichen. Der Spieler selbst kann das blaue Zeichen dirigieren, dabei kann dieses nicht diagonal und nicht auf schwarze Felder gehen. Die kleinen, schwarzen Punkte, die gezählt werden, kann man fressen. Geht man aber nocheinmal über das Feld, auf dem man vorher einen Punkt gefressen hat, färbt sich dieses Feld schwarz und kann nicht mehr betreten werden.
Haben Sie 350 Punkte gesammelt, öffnet sich die Schleuse zum grünen Punkt, der links auf dem Bildschirm ist. Nun kann man diesen grünen Punkt erreichen; worauf das nächste Bild erscheint, das blaue Zeichen rechts oben sichtbar wird und man nach links gehen muß.
Der anfangs erwähnte rote Kreis, welcher im Gegensatz zum blauen diagonal gehen kann, versucht sich demselben zu nähern bzw. diesen zu fangen. Wenn das letztere eingetreten ist, ist das Spiel beendet und die erreichte Punktzahl und der Rekord werden ausgedruckt. Dabei muß man beachten, daß der rote Kreis die schwarzen Felder nicht betreten kann.
Beim zweiten Bild, das sich im Aufbau vom ersten unterscheidet, muß man 150 Punkte erreichen, damit sich die Schleuse öffnet. Beim dritten Bild ist der Gang schon geöffnet und Sie brauchen Ihn "nur" zu erreichen. Hier kommt jedoch der rote Punkt ins Spiel, welcher dies sehr erschwert, da das Feld offen ist und man den roten Punkt schlecht einkreisen kann.
Gespielt wird mit Joystick!
Viel Spaß!

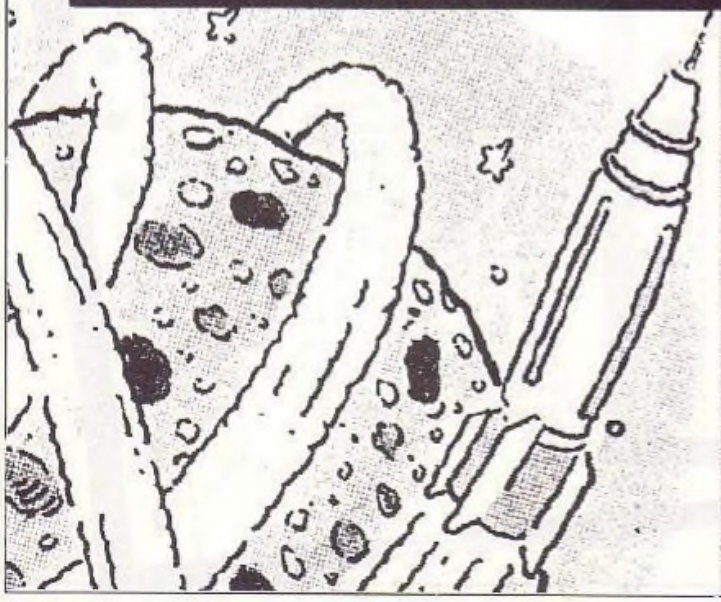
```
100 CALL CLEAR
110 CALL SCREEN(2)
120 CALL CHAR(120,"00183C24243C18")
130 CALL CHAR(128,"00000001")
140 CALL CHAR(136,"FFFFFFFFFFFFFFFF")
150 CALL CHAR(144,"00003C3C3C3C")
160 CALL CHAR(145,"00003C3C3C3C")
170 CALL CHAR(146,"00003C3C3C3C")
180 CALL CHAR(152,"0018245A5A2418")
190 CALL CHAR(153,"0018245A5A2418")
200 FOR I=2 TO 9
210 CALL COLOR(I,16,2)
220 NEXT I
230 CALL COLOR(12,7,16)
240 CALL COLOR(13,2,16)
250 CALL COLOR(14,16,16)
260 CALL COLOR(15,13,16)
270 CALL COLOR(16,5,16)
280 PRINT TAB(10);"WANDERING"::::::::::::
290 CALL SOUND(500,294,0,392,0,494,0)
300 CALL SOUND(250,294,0,392,0,494,0)
310 CALL SOUND(250,392,0,494,0,587,0)
320 CALL SOUND(1000,294,0,349,0,440,0)
330 PRINT ::::::::::::
340 FOR I=1 TO 100
350 NEXT I
360 CALL CLEAR
364 E=0
370 IF C<>146 THEN 390
380 E=E+500
390 Z=13
400 S=17
410 D=136
420 M=2
430 A=3
440 F=3
450 G=3
460 P=0
470 Q=250
480 FOR I=3 TO 23 STEP 2
490 CALL HCHAR(I,3,128,29)
500 NEXT I
510 FOR I=3 TO 31 STEP 2
520 CALL VCHAR(4,I,128,19)
530 NEXT I
540 CALL HCHAR(23,2,144)
550 CALL HCHAR(22,3,32)
560 CALL HCHAR(23,4,32)
570 CALL HCHAR(13,17,152)
580 GOSUB 1110
590 GOSUB 1370
600 GOTO 580
610 CALL CLEAR
620 E=E+P
630 P=0
640 Q=150
650 M=2
660 F=3
670 G=3
680 A=3
690 Z=3
700 S=32
710 FOR I=3 TO 23 STEP 10
720 CALL HCHAR(I,3,128,29)
730 NEXT I
740 FOR I=3 TO 31 STEP 7
750 CALL VCHAR(4,I,128,19)
760 NEXT I
770 FOR I=4 TO 30 STEP 2
780 FOR R=4 TO 22 STEP 2
790 CALL HCHAR(R,I,128)
800 NEXT R
810 NEXT I
820 FOR I=5 TO 29 STEP 2
830 FOR R=5 TO 21 STEP 2
840 CALL HCHAR(R,I,128)
850 NEXT R
860 NEXT I
870 CALL HCHAR(23,2,145)
880 CALL HCHAR(22,3,32)
890 CALL HCHAR(23,4,32)
900 CALL HCHAR(3,32,153)
910 GOSUB 1110
920 GOSUB 1370
930 GOTO 910
940 CALL CLEAR
950 D=32
960 E=E+P+250
970 M=2
980 F=3
990 G=3
1000 A=3
1010 Z=3
1020 S=32
1030 FOR I=3 TO 23
```

```
1040 CALL HCHAR(I,3,136,29)
1050 NEXT I
1060 CALL HCHAR(23,2,146)
1070 CALL HCHAR(3,32,153)
1080 GOSUB 1110
1090 GOSUB 1370
1100 GOTO 1080
1110 CALL JOYST(1,X,Y)
1120 IF (X+Y=0)+(X+Y=8)+(X+Y=-8)THEN 1360
1130 CALL GCHAR(Z-Y/4,S+X/4,C)
1140 IF C=32 THEN 1350
1150 IF C=136 THEN 1190
1160 B=152
1170 P=P+1
1180 GOTO 1200
1190 B=153
1200 CALL GCHAR(Z,S,V)
1210 IF V=153 THEN 1240
1220 N=136
1230 GOTO 1250
1240 N=32
1250 CALL HCHAR(Z,S,N)
1260 Z=Z-Y/4
1270 S=S+X/4
1280 CALL HCHAR(Z,S,B)
1290 IF C=120 THEN 1570
1300 IF P<Q THEN 1360
1310 IF P>Q THEN 1330
1320 CALL HCHAR(23,4,136)
1330 IF C=144 THEN 610
1340 IF C=145 THEN 940
1350 IF C=146 THEN 360
1360 RETURN
1370 IF M>Z THEN 1410
1380 IF M=Z THEN 1420
1390 M=M+1
1400 GOTO 1420
1410 M=M-1
```

```
1420 IF A>S THEN 1460
1430 IF A=S THEN 1470
1440 A=A+1
1450 GOTO 1470
1460 A=A-1
1470 CALL GCHAR(M,A,H)
1480 IF H=32 THEN 1540
1490 CALL HCHAR(F,G,C)
1500 CALL HCHAR(M,A,120)
1510 IF (H=152)+(H=153)THEN 1570
1520 F=M
1530 G=A
1540 M=F
1550 A=G
1560 RETURN
1570 FOR I=1 TO 15
1580 CALL HCHAR(Z,S,136)
1590 FOR L=I*220 TO (I+3)*220 STEP 110
1600 CALL SOUND(-500,L,15-I,4180-L,I)
1610 NEXT L
1620 CALL HCHAR(Z,S,120)
1630 NEXT I
1640 CALL CLEAR
1650 E=E+P
1660 PRINT :
1670 PRINT "DEINE PUNKTZAHL :";E;
1680 IF E<U THEN 1700
1690 U=E
1700 PRINT "WILLLST DU NOCH EINMAL"
1710 PRINT : SPIELEN ? (J)"
1720 CALL KEY(0,K,ST)
1730 IF ST=0 THEN 1720
1740 IF K=106 THEN 360
1750 CALL CLEAR
1760 FOR I=2 TO 9
1770 CALL COLOR(I,2,2)
1780 NEXT I
1790 PRINT "GESCHRIEBEN VON ROLF HEIKAMP"
1800 END
```

Moonpatrol

für den TI-99/4A Extended Basic



Mit diesem Spiel, welches eigentlich aus 5 einzelnen Spielen besteht, machen Sie eine Expedition auf dem Mond.

Es wurde Ihnen zur Aufgabe gemacht, mit einem Fahrzeug den Mond 3-mal zu umrunden.

Damit dies nicht so einfach zu bewältigen ist, werden Ihnen unterwegs einige Hindernisse und Gefahren begegnen, zu denen wir hier aber nicht mehr verraten wollen.

Gesteuert wird das Mondfahrzeug mit dem Joystick, dessen Funktion in jedem Spiel auf die erforderlichen Fähigkeiten des Fahrzeuges beschränkt sind.

```
10 KE=15 :: KV=30 :: MPC-3 :: HU=-20 :: HH=160
20 CALL CLEAR :: CALL SCREEN(8)
30 DISPLAY AT(1,10):"MOON-PATROL"
40 DISPLAY AT(5,10):"AUGUST 1983"
50 DISPLAY AT(6,10):"STEFAN FUNK"
60 DISPLAY AT(7,8):"AM WALDGRABEN 6"
70 DISPLAY AT(8,10):"6450 HANAU 9"
80 DISPLAY AT(9,5):"TEL.:(06181)-571227"
90 DISPLAY AT(15,1):"MOECHTEN SIE DIE"
100 DISPLAY AT(16,1):"SPIELANLEITUNG (J:N) ?" :: ACCEPT AT(16,28)BEEP VALIDATE("
NJnj")SIZE(1):AN$
110 IF AN$="N" OR AN$="n" THEN 230
120 CALL CLEAR :: DISPLAY AT(1,9):"**************" :: DISPLAY AT(2,9):"*MOON-PATR
OL*" :: DISPLAY AT(3,9):"**************"
130 DISPLAY AT(4,9):"**ANLEITUNG**" :: DISPLAY AT(5,9):"**************"
140 DISPLAY AT(7,1):"FAHREN SIE IHRE" :: DISPLAY AT(8,1):"MOON-PATROL-CAR" :: DI
SPLAY AT(9,1):"UM DEN FREMDEN PLANETEN,"
150 DISPLAY AT(10,1):"AUF DEM SIE" :: DISPLAY AT(11,1):"BELANDET SIND," :: DISPL
AY AT(12,1):"UM IHN ZU ERFORSCHEN."
160 DISPLAY AT(13,1):"GEHEN SIE DABEI JEDER" :: DISPLAY AT(14,1):"GEFAHR AUS DEM
 WEG."
170 DISPLAY AT(15,5):"S T E U E R U N G"
180 DISPLAY AT(16,1):"FERNBEDIENUNG NACH OBEN" :: DISPLAY AT(17,1):"SCHUSS NACH
OBEN"
190 DISPLAY AT(19,1):"FERNBEDIENUNG NACH UNTEN" :: DISPLAY AT(20,1):"HUEPFEN"
200 DISPLAY AT(22,1):"AKTIONSKNOPF DRUECKEN" :: DISPLAY AT(23,1):"SCHUSS NACH RE
CHTS;"
210 DISPLAY AT(24,1):"START DES SPIELS"
220 CALL KEY(1,A,B):: IF A<>18 THEN 220
230 RANDOMIZE
240 CALL CLEAR :: CALL SCREEN(2):: CALL MAGNIFY(4)
250 CALL CHAR(40,"0000000001010317 7F633FFF0863F/630000000C0003FFECF09CFEFF61CC9E
CC")
260 CALL CHAR(64,"0000C0000002020202BFF3F333E333FFF00002B107C1030101 0FFFCCCECCCFC
FF")
270 CALL CHAR(56,"0000000000001080 4A3002B1047079300000000000000004A012C90212:BCEEF
C6")
280 CALL CHAR(60,"0000000C0000000000000000001C1010300000000000000C0000000000000000
80")
290 CALL CHAR(44,"0000000000C0000000000000001F3B7EFF0000000000000000000000000FBFCCE
FF")
300 CALL CHAR(48,"0000100B04043FFFFFFF7F2444B4B48B0000844BEB90FCFFFFFFFE24221112
24")
310 CALL CHAR(52,"00000C0000000000000000000C0000000C0000000000003CFE040000000000000
00")
320 CALL CHAR(136,"003C7EFFFF00DBDB"):: CALL COLOR(14,9,1)
330 CALL CHAR(128,"003C7EFFFF00DBDB")
340 CALL COLOR(2,15,15,13,4,1)
350 FOR I=21 TO 24 :: CALL HCHAR(I,1,45,32):: NEXT I
360 CALL SPRITE(#1,64,16,129,20)
370 CALL SPRITE(#2,44,15,129,70)
380 CALL SPRITE(#3,44,15,129,184)
390 CALL MELODY
400 CALL SOUND(-100,-2,0,110,0):: CALL HCHAR(2,15,136):: CALL SOUND(200,131,30)
410 IF MPC>1 THEN CALL SOUND(-100,-2,0,110,0):: CALL HCHAR(2,17,128):: CALL SOUN
D(200,131,30)
420 IF MPC>2 THEN CALL SOUND(-100,-2,0,110,0):: CALL HCHAR(2,19,128)
430 FOR I=1 TO 300 :: NEXT I :: CALL SOUND(-500,262,0,330,0,392,0):: CALL PATTER
N(#1,40):: CALL COLOR(#1,7):: FOR I=1 TO 200 :: NEXT I
440 CALL MOTION(#2,0,HU,#3,0,HU):: X=129 :: Y=20
450 GOSUB 480
460 GOSUB 500
470 GOTO 450
480 CALL COINC(ALL,C):: IF C=-1 THEN 660
490 RETURN
500 Q=Q+1 :: IF Q=KE THEN GOSUB 650 ELSE IF Q=KV THEN GOSUB 850
510 SCH=SCH+1 :: IF SCH=50 THEN GOSUB 860 ELSE IF SCH=100 THEN 890
520 CALL JOYST(1,A,B):: IF B=-4 THEN 530 ELSE CALL KEY(1,N,M):: IF N=18 THEN 600
 ELSE RETURN
530 CALL SOUND(-300,-5,0):: PZ=PZ+10 :: X=100 :: CALL LOCATE(#1,X,Y)
540 IF HU=0 THEN 580
550 CALL POSITION(#2,N,M):: IF M<80 THEN 710
560 FOR II=1 TO HH/8 :: GOSUB 670
570 NEXT II :: GOTO 590
580 FOR I=1 TO HH :: NEXT I
590 X=129 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-300,-3,0):: RETURN
600 CALL SOUND(-100,-7,0):: CALL SPRITE(#4,52,5,129,45,0,30):: PZ=PZ-1
610 CALL POSITION(#4,H,W):: IF W<150 THEN GOSUB 480 ELSE 640
620 GOSUB 500
```

```
630 GOTO 610
640 CALL DELSPRITE(#4):: RETURN
650 NU=1 :: Z=INT(2*RND)+1 :: Z=Z+1 :: CALL COLOR(#Z,13):: CALL PATTERN(#Z,4E)::
 RETURN
660 IF NU=0 THEN 710 ELSE 700
670 CALL COINC(ALL,C):: IF C=0 THEN RETURN
680 PZ=PZ+INT(PZ/10):: CALL DELSPRITE(#4):: CALL SOUND(-100,131,0,247,0,330,0,-7
,0)
690 FOR I=2 TO 3 :: CALL PATTERN(#I,44):: CALL COLOR(#I,15):: NEXT I :: NU=0 ::
RETURN
700 EE=0 :: FOR I=2 TO 3 :: CALL COINC(#4,#I,100,C):: EE=EE+C :: NEXT I :: IF EE
<0 THEN PZ=PZ+INT(PZ/10):: CALL SOUND(-100,131,0,247,0,330,0,-7,0):: GOTO 730
710 GOSUB 740
720 Q=0 :: HH=160 :: NU=0 :: SCH=0 :: KE=15 :: KV=30 :: HU=-20 :: GOTO 360
730 CALL DELSPRITE(#4):: CALL COLOR(#Z,15):: CALL PATTERN(#Z,44):: Q=0 :: NU=0 :
: GOTO 500
740 FOR I=1 TO 3 :: CALL SOUND(-100,-7,0):: NEXT I :: CALL SOUND(-100,-5,0):: CA
LL SOUND(-100,-7,0):: CALL SOUND(-100,-7,0)
750 CALL HCHAR(2,15,32,6)
760 CALL PATTERN(#1,56):: FOR I=2 TO 4 :: CALL MOTION(#I,0,0):: NEXT I :: FOR P=
1 TO 100 :: NEXT P :: CALL DELSPRITE(#1):: FOR I=1 TO 500 :: NEXT I :: CALL DELS
PRITE(ALL)
770 MPC=MPC-1 :: IF MPC<1 THEN 780 ELSE RETURN
780 CALL SOUND(500,196,0):: CALL SOUND(500,196,0):: CALL SOUND(100,196,0):: CALL
 SOUND(500,196,0)
790 CALL SOUND(500,233,0):: CALL SOUND(100,220,0):: CALL SOUND(250,220,0)
800 CALL SOUND(250,196,0):: CALL SOUND(250,196,0)
810 CALL SOUND(250,185,0):: CALL SOUND(1000,196,0)
820 CALL CLEAR :: CALL SCREEN(11):: CALL CHARSET :: DISPLAY AT(5,9):"AUSWERTUNG"
830 DISPLAY AT(2,5):"M O O N - P A T R O L" :: DISPLAY AT(15,7):PZ:"PUNKTE"
 :: DISPLAY AT(12,7):"SIE ERREICHTEN"
840 CALL KEY(1,A,B):: IF A=18 THEN RUN ELSE 840
850 NU=0 :: Q=0 :: CALL PATTERN(#Z,44):: CALL COLOR(#Z,15):: RETURN
860 CALL SOUND(500,262,0):: CALL SOUND(500,392,0):: Q=0 :: NU=0 :: PZ=PZ*2 :: KE
=5 :: KV=20 :: HU=-25 :: HH=110 :: CALL MOTION(#2,0,0,#3,0,0)
870 CALL SCREEN(4):: CALL DELSPRITE(#4):: FOR I=1 TO 50 :: NEXT I :: CALL SCREEN
(2)
880 CALL SPRITE(#2,44,15,129,70):: CALL SPRITE(#3,44,15,129,184):: FOR I=1 TO 20
0 :: NEXT I :: CALL MOTION(#2,0,HU,#3,0,HU):: RETURN
890 CALL MOTION(#2,0,0,#3,0,0)
900 CALL DELSPRITE(#4):: FOR I=1 TO 10 :: CALL SOUND(-100,131,0):: CALL SCREEN(4
):: CALL SOUND(-100,262,0):: NEXT I
910 CALL SOUND(500,262,0):: CALL SOUND(500,392,0):: PZ=PZ*2
920 CALL SCREEN(2):: SCH=0 :: NU=0 :: KSCH=0
930 FOR I=2 TO 3 :: CALL DELSPRITE(#I):: NEXT I :: CALL CHAR(44,"000000000000000
0000000600331780000000000000000000000006C0831E00")
940 FOR I=1 TO 100 :: NEXT I :: MSH=INT(10*RND)+1 :: MSH=MSH-40
950 CALL SPRITE(#1,40,7,X,Y):: CALL SPRITE(#2,44,8,129,100):: CALL SPRITE(#3,44,
8,129,190)
960 CALL SOUND(-500,131,0,262,0)
970 CALL MOTION(#2,0,MSH,#3,0,MSH)
980 CALL COINC(ALL,C):: IF C<0 THEN 1060
990 SCH=SCH+1 :: IF SCH=25 THEN 1040 ELSE CALL JOYST(1,A,B):: IF B=-4 THEN 1000
ELSE 980
1000 X=100 :: CALL SOUND(-1000,-6,0):: CALL LOCATE(#1,X,Y):: PZ=PZ+10
1010 FOR I=1 TO 80 :: NEXT I
1020 CALL SOUND(-100,131,0)
1030 X=129 :: CALL LOCATE(#1,X,Y)
1040 KSCH=KSCH+1 :: IF KSCH=25 THEN 1090
1050 SCH=0 :: MSH=INT(10*RND)+1 :: MSH=MSH-40 :: CALL MOTION(#2,0,MSH,#3,0,MSH):
: GOTO 980
1060 FOR I=2 TO 3 :: CALL MOTION(#I,0,0):: NEXT I :: FOR J=1 TO 50 :: NEXT J
1070 FOR I=2 TO 3 :: CALL DELSPRITE(#I):: NEXT I
1080 CALL CHAR(44,"000000000000000000000000001F3B7EFF000000000000000000000000F8FCC
EFF"):: GOTO 710
1090 CALL MOTION(#2,0,0,#3,0,0):: CALL SCREEN(4):: FOR I=1 TO 3 :: FOR S=-3 TO -
1 :: CALL SOUND(-100,S,0):: NEXT S :: NEXT I
1100 FOR I=1 TO 50 :: NEXT I :: CALL SOUND(500,262,0):: CALL SOUND(500,392,0)::
PZ=PZ*2
1110 CALL SCREEN(2)
1120 CALL DELSPRITE(#2):: CALL DELSPRITE(#3)
1130 CALL CHAR(44,"000000000000000000000000003F1F0F0300000000000000000000000000FCF0C
080")
1140 CALL CHAR(108,"000000000000000000000070C7FFF3F0C000000000000000000E043FFFEFCFC
FE3F")
1150 SCH=0 :: Q=0 :: DD=20 :: CALTE(#3,108,11,100,180):: CALL SPRITE(#2,44
,2,137,190)
1160 CALL SOUND(-500,131,0,262,0)
1170 CALL MOTION(#3,0,-55,#2,0,-20)
1180 Q=Q+1 :: IF Q=20 THEN GOSUB 1290
```

```
1190 DD=DD+1 :: IF DD=20 THEN CALL SPRITE(#3,108,11,100,180,0,-55)
1200 CALL POSITION(#2,K,L):: IF L<30 AND K>15 THEN 1310
1210 CALL JOYST(1,A,B):: IF B=-4 THEN 1220 ELSE IF B=4 THEN 1260 ELSE 1180
1220 X=100 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-500,-6,0)
1230 PZ=PZ+10 :: FOR I=1 TO 20 :: CALL COINC(ALL,C):: IF C=-1 THEN 1310
1240 NEXT I
1250 X=129 :: CALL LOCATE(#1,X,Y):: CALL SOUND(-500,131,0):: GOTO 1180
1260 PZ=PZ-1 :: CALL SPRITE(#4,60,14,100,20):: CALL SOUND(-100,-7,0,131,0):: CAL
L POSITION(#3,K,L):: IF L<80 THEN 1270 ELSE 1280
1270 CALL DELSPRITE(#4):: CALL DELSPRITE(#3):: CALL SOUND(-100,-7,0,1480,0,220,0
,622,0):: DD=0 :: PZ=PZ+500 :: GOTO 1180
1280 CALL SOUND(-100,-1,0):: CALL DELSPRITE(#4):: GOTO 1180
1290 SCH=SCH+1 :: IF SCH=5 THEN 1330 ELSE NM=INT(100*RND)+1 :: NM=NM+90 :: CALL
SOUND(-100,-1,0,262,0)
1300 CALL SPRITE(#2,44,2,137,NM,0,-25):: RETURN
1310 CALL MOTION(#2,0,0,#3,0,0):: CALL DELSPRITE(#4):: CALL DELSPRITE(#2):: X=14
0 :: CALL LOCATE(#1,X,Y):: X=129
1320 CALL SOUND(-100,-7,0,131,0,220,0,110,0):: GOTO 1080
1330 CALL MOTION(#2,0,0,#3,0,0):: CALL SCREEN(4):: FOR I=1 TO 5 :: CALL SOUND(-1
00,131,0):: CALL SOUND(-100,262,0):: NEXT I
1340 CALL SCREEN(2):: FOR J=1 TO 50 :: NEXT J :: PZ=PZ*2 :: CALL SOUND(500,262,0
):: CALL SOUND(500,392,0)
1350 FOR I=2 TO 4 :: CALL DELSPRITE(#I):: NEXT I
1360 CALL CHAR(44,"000000000000000000C000000C61610C00000C000000000000000000030B68
630")
1370 CALL SPRITE(#2,44,8,127,1)
1380 CALL SPRITE(#3,44,6,129,80)
1390 CALL SPRITE(#4,44,5,129,170)
1400 CALL SOUND(-500,262,0,131,0)
1410 SCH=0 :: Q=0
1420 FOR I=1 TO 100 :: NEXT I :: FOR I=2 TO 4 :: CALL MOTION(#I,0,-30):: NEXT I
1430 CALL JOYST(1,A,B):: IF B=-4 THEN 1470
1440 Q=Q+1 :: IF Q=5 THEN GOSUB 1500
1450 CALL COINC(ALL,C):: IF C=-1 THEN 1510
1460 CALL JOYST(1,A,B):: IF B=-4 THEN 1470 ELSE 1440
1470 X=100 :: CALL SOUND(-500,-6,0):: CALL LOCATE(#1,X,Y)
1480 PZ=PZ+10 :: FOR I=1 TO 75 :: NEXT I
1490 X=129 :: CALL SOUND(-500,-3,0):: CALL LOCATE(#1,X,Y):: GOTO 1440
1500 MSH=INT(10*RND)+1 :: MSH=MSH-35 :: CALL MOTION(#2,0,MSH,#3,0,MSH,#4,0,MSH):
: Q=0 :: SCH=SCH+15 :: IF SCH=1 THEN 1520 ELSE RETURN
1510 CALL MOTION(#2,0,0,#3,0,0,#4,0,0):: CALL SOUND(-100,-7,0,262,0):: FOR I=2 T
O 4 :: CALL DELSPRITE(#I):: NEXT I :: GOTO 1080
1520 FOR I=2 TO 4 :: CALL MOTION(#I,0,0):: NEXT I :: CALL SCREEN(4):: FOR I=1 TO
5 :: CALL SOUND(-100,131,0):: CALL SOUND(-100,262,0):: CALL SOUND(-100,523,0):: 
NEXT I
1530 FOR J=1 TO 50 :: NEXT J :: PZ=PZ*2 :: CALL SOUND(500,262,0):: CALL SOUND(50
0,392,0)
1540 CALL SCREEN(2)
1550 FOR S=2 TO 4 :: CALL DELSPRITE(#S):: NEXT S
1560 CALL SPRITE(#2,64,8,129,192,0,-10)
1570 CALL POSITION(#2,X,Y):: CALL SOUND(-300,-1,0,131,0,220,0):: IF Y>30 THEN 15
70
1580 CALL SOUND(-500,131,0,262,0,523,0):: CALL MOTION(#2,0,0):: FOR I=1 TO 300 :
: NEXT I
1590 CALL SOUND(250,262,0,330,0)
1600 CALL SOUND(250,294,0,349,0)
1610 CALL SOUND(500,330,0,392,0)
1620 CALL SOUND(500,262,0,330,0)
1630 CALL SOUND(500,349,0,440,0)
1640 CALL SOUND(1000,330,0,392,0)
1650 FOR I=1 TO 200 :: NEXT I :: CALL HCHAR(2,15,32,6)
1660 MPC=MPC+1 :: IF MPC>3 THEN MPC=3
1670 CALL DELSPRITE(ALL):: CALL CHAR(44,"000000000000C000000000001F3B7EFF0000000
00000000000000C00F8FCCEFF"):: GOTO 720
1680 SUB MELODY
1690 FOR I=1 TO 300 :: NEXT I
1700 FOR MEL=1 TO 3
1710 CALL SOUND(200,130,0,-1,10)
1720 CALL SOUND(200,130,0,-1,10)
1730 CALL SOUND(200,195,0,-1,10)
1740 CALL SOUND(100,130,0,-1,10)
1750 CALL SOUND(250,130,0,-1,10)
1760 CALL SOUND(100,130,0,-1,10)
1770 CALL SOUND(250,130,0,-1,10)
1780 CALL SOUND(200,195,0,-1,10)
1790 CALL SOUND(200,195,0,-1,10)
1800 NEXT MEL
1810 CALL SOUND(200,130,0,-1,10)
1820 FOR I=1 TO 300 :: NEXT I
1830 SUBEND
```

REVIEWS

Atic Atac
für den Spectrum 48K

Ultimate hat uns schon eine ganze Reihe von hervorragenden Programmen für den Spectrum beschert. Mit Atic Atac, in England zur Zeit in den Hitlisten ganz vorn, wird diese Tradition fortgesetzt.
Es geht darum, den goldenen Schlüssel zu finden, der in einem geheimnisvollen Ort irgendwo versteckt ist.

Das Schloß beinhaltet eine große Anzahl von Räumen, die jeweils ein bis drei Türen haben. Nach jedem Durchwandern eines Raumes wird ein neues Bild gezeichnet. Das Programm arbeitet sehr schnell und benötigt fast keine Pause zum Ersteller eines neuen Bildes.
Die Qualität der Zeichnungen in diesem Spiel ist eine der besten, die wir jemals auf einem Spectrum gesehen haben. Die Bewegung der Figuren ist sehr ansprechend und man hat recht originelle Charaktere, wie beispielsweise Hexen mit einem Besen und andere Fabelfiguren, kreiert.
Kontrolliert wird das Programm entweder mittels eines Kempston oder AGF-Joysticks oder über die vier Cursortasten. Allerdings ist der einzig wahre Weg, das Spiel wirklich zu beherrschen, nur mittels Joystick möglich.
Dieses Programm dürfte sicherlich das beste sein, was Ultimate bis jetzt programmiert hat und kann allen Spectrum besitzern wärmstens empfohlen werden.

Gute Nachricht für Willy-Fans

Kumpel Willy, der Held von Bug Bytes Bestsellerprogramm "Manic Miner", wird in Kürze auch auf dem Commodore 64 seine Künste zeigen.
Der Autor der Spectrumversion Matthew Smith, der sich mit seiner Firma Software Projects selbstständig gemacht hat, hat die Vermarktung dieses Spieles angekündigt.
Wie uns mitgeteilt wurde, soll ein Superpreis für denjenigen ausgesetzt werden, der all die verschiedenen Stationen dieses Programms als erster bewältigt.
Nachdem schon einige andere Umsetzungen von Spectrum Programmen auf den Commodore 64 recht erfolgreich waren, wird auch dieses sicher wieder ein ausgezeichnetes Spiel werden.
Eine weitere gute Nachricht für alle Willy-Fans: Manic Miner Teil II ist bereits in Vorbereitung.

Korrektur Star Wars (VC-20) HC 2/84

Zum Eintippen des Programmes: Bei Verwendung des Zeichensatzes aus der HC 2/84 muß folgendermaßen vorgegangen werden:
1. POKE 44,24:POKE 6144,0:NEW
2. Zeichensatz (HC 2/84 Seite 30) eingeben
3. RUN
4. POKE 44,24:POKE 6144,0:NEW
5. Hauptprogramm eingeben
6. RUN
Zum Abspeichern des Programmes
POKE 44,18
eingeben, dann kann das Programm auf Diskette oder Kassette gesaved werden.
Später wird es nur mit Load eingeladen und mit RUN gestartet.

Berichtigung zu Antares (TI-99/4A) aus HC 3/84

Folgende Zeilen müssen in das Listing eingefügt werden:
2740 GOTO 2810
2750 REM

Amor

für den TI-99/4A (Extended Basic)

Als Amor, der Gott der Liebe, wollen Sie die Schöne auf der Schaukel erobern. Sie müssen die Herzen mit Ihrem Liebespfeil abschießen. Sind alle Herzen getroffen, haben Sie ihre Zuneigung gewonnen. Eine Belohnung winkt. Durch Schußwinkeleingabe (1-24) können die Pfeile gesteuert werden. Insgesamt stehen 8 Pfeile zur Verfügung.

Ablauf:
5-100 Zeichendefinierung
105-165 Vorspann
190-215 Erklärung
225 Variablen für 1. Musik
230-385 1. Musik
390-555 Grafik
560-715 Hauptprogramm
755 Variablen für 2. Musik
760-940 2. Musik
945 Ende Schluß
#1-#5 Herzen
#17,#18 Herzen
#9,#10,#11,#14 Wolken
#8 Mann
#7 Frau
#13 Pfeil
#20 Frau Tanz
#21 Mann Tanz
Pun = Punkte
PF = Pfeile
500-595 Bewegung des Pfeils

```
5 CALL DELSPRITE(ALL):: CALL CLEAR
10 PUN=0
15 H$="1C3E7FFFFFFFFFFF7F3F1F8F07030101387CFEFFFFFFFFFFEFCF8F0E0C08080" :: CALL
 CHAR(96,H$)
20 M$="E0C1A2140814121313140803030202020C869F89C5A69C9F0FFF3F7EFC18387EE3" :: CALL
 CHAR(100,M$)
25 MM$="05030201003FC3010103030704040C0000C040CC88F8C0C0C0E0E0202020233" :: CAL
L CHAR(36,MM$)
30 F$="0303878911478D87C3733F1F3000030001189D1A205C971C1C9E6FE909090D8" :: CALL
 CHAR(104,F$)
35 FF$="01330D0031120F0103030707020202CC080C080F0A0C0C1C0E0E3F010101000" :: CAL
L CHAR(48,FF$)
40 GW$="0E3F7EFFFF7F7E3E3E3F3F1F0F0707C0F0F8FEFFFF732303078EDFFFFEFCC0" :: CAL
L CHAR(108,GW$)
45 KW$="0F7F7FF3F0F0787C7E7F7F3F0F030000C0FEFF9F1F1F3F7FFFFFFFFFFFC78000" :: CAL
L CHAR(112,KW$)
50 KW1$="030FFFFFFFFFFFFFFFFFFFF0E0C06000EF0E0E0F0F0FEFFFFFF0F0E04000000" :: CA
LL CHAR(132,KW1$)
55 MS$="0107C221100F0301010303010101030000C0404000D0D0D3C0EC2060404040C0" :: CAL
L CHAR(116,MS$)
60 FS$="0103050031020703908130707010101C080C688F0A0C0C0C0E0E0604040600" :: CAL
L CHAR(120,FS$)
65 CALL MAGNIFY(3)
70 CALL CLEAR
75 P$="E0C0A0100004D2010000000000000000000000000000000000000000000000000" :: CAL
L CHAR(124,P$)
80 CALL CHAR(94,"FFFFFFFFFFFFFFFF")
85 CALL CHAR(81,"FFFFFFFFFFFFFFFF")
90 CALL CHAR(33,"FFFFFFFFFFFFFFFF")
95 CALL CHAR(34,"0000000000000000")
100 CALL CHAR(35,"0101010101010101")
105 CALL CLEAR :: CALL DELSPRITE(ALL)
110 M$="    AMOR"                                J. UFERMANN"
115 AUT$="                PROGRAMMIERT:          W. UFERMANN"
120 ID$="                IDEE          :
125 CALL SCREEN(2):: CALL MAGNIFY(4)
130 CALL COLOR(2,7,2,4,7,2)
135 CALL CHAR(136,RPT$("99",16)&RPT$("99",16)):: CALL CHAR(127,"3C4299A1A199423C
")
140 FOR N=5 TO 12 :: CALL COLOR(N,14,1):: NEXT N :: DISPLAY AT(15,16):CHR$(127)&
"PROGRAMM  BY" :: DISPLAY AT(17,16):AUT$
145 DISPLAY AT(20,15):ID$
150 DISPLAY AT(8,16):M$
155 FOR N=1 TO 25 :: CALL SPRITE(#N,136,1,230,68+35*SIN(N/4),8,8):: NEXT N
160 FOR N=1 TO 25 :: CALL MOTION(#N,35,8):: CALL COLOR(#N,INT(N/2)+3):: NEXT N
165 FOR I=1 TO 500  : NEXT I
170 PRINT "   ANWEISUNG WUENSCH? (J/N) "
175 ACCEPT AT(23,28)BEEP VALIDATE("JN"):R$ :: IF R$="N" THEN 395
180 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL SCREEN(2)
185 CALL COLOR(3,14,2)
190 PRINT " -AMOR-AMOR-AMOR-AMOR-AMOR-" :: PRINT
195 PRINT "- SIE SIND AMOR, DER GOTT      DER LIEBE UND WOLLEN DIE     SCHOENE AUF
 DER SCHAUKEL     EROBERN." :: PRINT
200 PRINT "- DAZU MUESSEN SIE ALLE        HERZEN MIT IHREM LIEBES-     PFEIL ABSCH
IESSEN. WENN      ALLE HERZEN GETROFFEN      SIND,HABEN SIE"
205 PRINT "  IHRE ZUNEIGUNG GEWONNEN." :: PRINT :
210 PRINT "  --EINE BELOHNUNG WINKT--" :: PRINT :
215 PRINT "- DURCH SCHUSSWINKEL-EINGABE  (1-24) KOENNEN SIE IHREN     PFEIL STEUE
RN."
220 PRINT :: PRINT "          VIEL GLUECK"
```

```
225 VV=200 :: H=400 :: DV=600 :: TC=131 :: TH=123 :: TE=165 :: TG=196 :: TF=175
:: EE=330 :: FF=349 :: GG=392 :: CC=262 :: DD=294 :: XX=1
230 FOR I=1 TO 2
235 CALL SOUND(H,TC,0,EE,0)
240 CALL SOUND(VV,TE,0)
245 CALL SOUND(VV,TG,0,EE,0)
250 CALL SOUND(H,TC,0,GG,0)
255 CALL SOUND(VV,TE,0)
260 CALL SOUND(VV,TG,0)
265 CALL SOUND(H,TC,0,CC,0)
270 CALL SOUND(VV,TE,0)
275 CALL SOUND(VV,TG,0,DD,0)
280 CALL SOUND(H,TC,0,EE,0)
285 CALL SOUND(VV,TE,0)
290 CALL SOUND(VV,TG,0)
295 CALL SOUND(H,TH,0,DD,0)
300 CALL SOUND(VV,TF,0)
305 CALL SOUND(VV,TG,0,DD,0)
310 CALL SOUND(H,TH,0,GG,0)
315 CALL SOUND(VV,TF,0)
320 CALL SOUND(VV,TG,0,GG,0)
325 IF XX=2 THEN 370 ELSE 330
330 CALL SOUND(H,TC,0,EE,0)
335 CALL SOUND(VV,TE,0)
340 CALL SOUND(VV,TG,0,DD,0)
345 CALL SOUND(H,TC,0,CC,0)
350 CALL SOUND(VV,TE,0)
355 CALL SOUND(VV,TG,0)
360 XX=XX+1
365 NEXT I
370 CALL SOUND(H,TC,0,CC,0)
375 CALL SOUND(VV,TE,0)
380 CALL SOUND(VV,TG,0,EE,0)
385 CALL SOUND(DV,TC,0,CC,0):: CALL CLEAR :: DISPLAY AT(12,5)SIZE(22):"PS. SIE H
ABEN 8 PFEILE"
386 FOR I=1 TO 800 :: NEXT I
390 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL SCREEN(5)
395 CALL CHAR(127,"")
400 CALL COLOR(1,15,1,7,2,1,8,3,3)
405 CALL MAGNIFY(3)
410 CALL HCHAR(23,1,81,32)
415 CALL HCHAR(22,10,81,9)
420 CALL HCHAR(22,30,81,3)
425 CALL HCHAR(21,10,33,6)
430 CALL VCHAR(11,14,33,11)
435 CALL VCHAR(11,13,33,11)
440 CALL VCHAR(17,12,33,5)
445 CALL HCHAR(20,11,33)
450 CALL HCHAR(14,12,33)
455 CALL VCHAR(11,12,33)
460 CALL HCHAR(13,9,33,3)
465 CALL HCHAR(10,11,94,4)
470 CALL HCHAR(9,11,94,5)
475 CALL HCHAR(8,9,94,8)
480 CALL HCHAR(7,9,94,14)
485 CALL HCHAR(6,8,94,18)
490 CALL HCHAR(5,8,94,18)
495 CALL HCHAR(4,1,94,31)
500 CALL HCHAR(1,1,94,95)
505 CALL HCHAR(2,1,94,32)
510 CALL HCHAR(3,1,94,31):: CALL HCHAR(20,17,33,2)
515 CALL SPRITE(#1,96,7,50,80,0,4,#2,96,7,16,72,0,4.4,#3,96,7,32,112,0,3.9,#4,96
,7,24,152,0,4.9)
520 CALL SPRITE(#5,96,7,24,216,0,4,#7,104,16,146,129,#9,108,16,64,100,0,2,#10,11
2,16,73,200,0,2.5)
525 CALL SPRITE(#11,132,16,73,215,0,2.5,#14,108,16,60,150,0,1.4,#8,100,16,152,23
2,#16,112,16,48,1,0,1)
530 CALL SPRITE(#15,132,16,48,16,0,1,#17,96,7,8,184,0,5,#18,96,7,84,88,0,5)
535 CALL VCHAR(8,17,34,12):: CALL HCHAR(2,5,33):: CALL HCHAR(3,6,33)
540 CALL HCHAR(2,22,33,5):: CALL HCHAR(3,20,33,2):: CALL HCHAR(4,18,33,2):: CALL
HCHAR(5,17,33)
545 CALL VCHAR(8,18,35,12):: CALL HCHAR(1,17,33):: CALL HCHAR(2,16,33):: CALL HC
HAR(3,15,33)
550 CALL HCHAR(24,5,81,28):: CALL HCHAR(6,11,33):: CALL HCHAR(7,12,33):: CALL HC
HAR(8,13,33)
555 CALL COLOR(4,2,1,5,2,1):: PF=8
560 DISPLAY AT(1,1):"PUNKTE: ";PUN :: DISPLAY AT(1,19):"PFEILE:";PF :: IF PUN>=3
50 THEN 740
565 CALL DELSPRITE(#13):: DISPLAY AT(24,1):"WINKEL: " :: CALL SPRITE(#13,124,2,1
52,232)
570 IF PF<1 THEN 945 :: ACCEPT AT(24,8)SIZE(2)BEEP VALIDATE(DIGIT):W
571 IF W>24 THEN 570
575 PF=PF-1 :: CALL HCHAR(24,1,81,32)
580 V0=47
585 X=-70 :: T=0
590 V=V0+X*T
595 CALL MOTION(#13,-V,-W)
600 IF -V>V0 THEN 560
605 CALL COINC(#13,#1,13,A)
610 IF A=-1 THEN 615 ELSE 620
615 CALL DELSPRITE(#13):: CALL SOUND(100,-6,0):: CALL DELSPRITE(#1):: PUN=PUN+50
:: GOTO 560
```

Amor

```
620 CALL COINC(#13,#2,13,B)
625 IF B=-1 THEN 630 ELSE 635
630 CALL DELSPRITE(#13):: CALL SOUND(100,-6,0):: CALL DELSPRITE(#2):: PUN=PUN+50
 :: GOTO 560
635 CALL COINC(#13,#3,13,C)
640 IF C=-1 THEN 645 ELSE 650
645 CALL DELSPRITE(#3):: CALL SOUND(100,-6,0):: PUN=PUN+50 :: CALL DELSPRITE(#13
):: GOTO 560
650 CALL COINC(#13,#4,13,D)
655 IF D=-1 THEN 660 ELSE 665
660 CALL DELSPRITE(#4):: CALL DELSPRITE(#13):: CALL SOUND(100,-6,0):: PUN=PUN+50
 :: GOTO 560
665 CALL COINC(#13,#5,13,E)
670 IF E=-1 THEN 675 ELSE 680
675 CALL DELSPRITE(#5):: CALL DELSPRITE(#13):: CALL SOUND(100,-6,0):: PUN=PUN+50
 :: GOTO 560
680 CALL COINC(#13,#17,13,F)
685 IF F=-1 THEN 690 ELSE 695
690 CALL DELSPRITE(#17):: CALL DELSPRITE(#13):: CALL SOUND(100,-6,0):: PUN=PUN+5
0 :: GOTO 560
695 CALL COINC(#13,#18,13,G)
700 IF G=-1 THEN 705 ELSE 710
705 CALL DELSPRITE(#13):: CALL DELSPRITE(#18):: CALL SOUND(100,-6,0):: PUN=PUN+5
0 :: GOTO 560
710 CALL COINC(#13,#2,13,H):: IF H=-1 THEN 715 ELSE 720
715 CALL DELSPRITE(#13):: CALL DELSPRITE(#2):: CALL SOUND(100,-6,0):: PUN=PUN+50
 :: GOTO 560
720 REM
725 IF -V>VO THEN CALL DELSPRITE(#13):: GOTO 560
730 T=T+.2
735 GOTO 590
740 CALL DELSPRITE(#1,#2,#3,#4,#5,#7,#8,#13,#18,#17)
745 CALL SPRITE(#20,120,16,160,176,#21,116,16,160,192)
750 CALL SPRITE(#21,116,16,160,192)
755 GA=1150 :: DR=900 :: HA=750 :: VI=400 :: AC=275
760 CALL SOUND(DR,196,2,156,2)
765 CALL SOUND(25,14000,30)
770 CALL SOUND(DR,196,2,156,2):: CALL PATTERN(#20,40,#21,36)
775 CALL SOUND(25,14000,30)
780 CALL SOUND(AC,196,2,156,2)
785 CALL SOUND(AC,175,2,147,2):: CALL PATTERN(#20,120,#21,116)
790 CALL SOUND(AC,196,2,156,2)
795 CALL SOUND(500,233,2,196,2)
800 CALL SOUND(AC,208,2,175,2):: CALL PATTERN(#20,40,#21,36)
805 CALL SOUND(DR,175,2,147,2)
810 CALL SOUND(DR,175,2,147,2)
815 CALL SOUND(AC,175,2,147,2):: CALL PATTERN(#20,120,#21,116)
820 CALL SOUND(AC,233,2,147,2)
825 CALL SOUND(AC,208,2,147,2)
830 CALL SOUND(VI,196,2,156,2):: CALL PATTERN(#20,40,#21,36)
835 CALL SOUND(AC,14000,30)
840 CALL SOUND(DR,196,2,156,2)
845 CALL SOUND(25,14000,30):: CALL PATTERN(#20,120,#21,116)
850 CALL SOUND(DR,196,2,156,2)
855 CALL SOUND(25,14000,30):: CALL SOUND(AC,196,2,156,2):: CALL PATTERN(#20,40,#
21,36)
860 CALL SOUND(AC,175,2,147,2)
865 CALL SOUND(AC,196,2,156,2):: CALL SOUND(VI,233,2,196,2):: CALL PATTERN(#20,1
20,#21,116)
870 CALL SOUND(AC,208,2,175,2)
875 CALL SOUND(AC,175,2,147,2):: CALL SOUND(AC,196,2,156,2):: CALL PATTERN(#20,4
0,#21,36)
880 CALL SOUND(AC,208,2,175,2)
885 CALL SOUND(AC,262,2,175,2):: CALL SOUND(AC,233,2,196,2):: CALL PATTERN(#20,1
20,#21,116)
890 CALL SOUND(AC,208,2,156,2)
895 CALL SOUND(DR,196,2,156,2):: CALL SOUND(VI,196,2,156,2):: CALL PATTERN(#20,4
0,#21,36)
900 CALL SOUND(AC,14000,30)
905 CALL SOUND(DR,262,2,156,2):: CALL SOUND(DR,233,2,156,2):: CALL PATTERN(#20,1
20,#21,116)
910 CALL SOUND(DR,294,2,262,2,175,2)
915 CALL SOUND(DR,311,2,233,2,196,2):: CALL SOUND(AC,175,2):: CALL PATTERN(#20,4
0,#21,36)
920 CALL SOUND(AC,196,2)
925 CALL SOUND(AC,208,2):: CALL SOUND(AC,262,2):: CALL PATTERN(#20,120,#21,116)
930 CALL SOUND(AC,233,2):: CALL SOUND(AC,147,2)
935 CALL SOUND(DR,156,2)
940 CALL SOUND(VI,156,2)
945 CALL DELSPRITE(#13):: DISPLAY AT(13,19)SIZE(9)BEEP:"GAME OVER" :: DISPLAY AT
(15,20):"<TASTE>"
946 DISPLAY AT(13,19)SIZE(9):"         "
950 CALL KEY(0,K,S)
955 IF S=0 THEN 945
960 GOTO 5
```

Zeichen und Sprites definieren

für den TI-99/4A (Extended Basic)

Das zu konstruierende Zeichen wird in vergrößerter Form auf dem Bildschirm entworfen und dargestellt. Anschliessend wird der HEX-Code des Zeichens berechnet und angezeigt. Es können maximal vier in einem Block zusammengefaßte Zeichen entwickelt werden.

Nach dem Start des Programms mit < RUN> erscheint das Hauptmenue mit der Frage, ob ein Zeichen entwickelt oder ein bereits entwickeltes Zeichen dargestellt werden soll.

Nach dem Drücken der Taste < 1> kann ein Zeichen entwickelt werden. Wiederrum kann mit der Taste < 1> der Cursor (Pixel) eingeschaltet und mit der Taste < 0> ausgeschaltet werden. Der Cursor wird mit den Tasten < E> OBEN, < X> UNTEN, < S> LINKS, < D> RECHTS gesteuert. Durch Drücken der Taste < 5> wird der HEX-Code errechnet (Dauer ca. 20 Sek.). Mit der Taste < M> kann der Vergrößerungsfaktor (Magnify) von 1 bis 4 eingestellt werden. Durch Drücken der Taste < C> wird die Farbe des entwickelten Zeichens und durch Taste < B> die Hintergrundfarbe (Screen) von 1-16 eingestellt.

Mit < T> kann das entwickelte Zeichen nochmals geändert werden, oder durch < A> ganz gelöscht. Durch Drücken der Taste < Q> kehrt man zum Hauptmenue zurück. Mit < L> wird der HEX-Code als gesamtes und auch für jedes Feld einzeln dargestellt. Weiterhin kann die ausgewählte Screen- und Color-Farbe abgelesen werden. Durch die Taste < D> besteht die Möglichkeit, den erzeugten HEX-Code auf einem EPSON RX 80 auszudrucken und durch < Q> kehrt man zum Hauptmenue zurück.

Durch die Taste < 2> gelangt man nun vom Hauptmenue zum zweiten Teil des Programms, mit dem die Möglichkeit besteht, einen bereits vorhandenen Zeichencode zu überprüfen.

Es stehen 4 Eingabezeilen (a 16 Zeichen) zur Verfügung. Wenn eine Eingabezeile nicht ganz benötigt wird, muß der Rest der Zeile mit Nullen aufgefüllt werden. Mit ENTER wird der Eingabevorgang für jede Zeile beendet, und die Tasten < B, C, L, M, Q, T> können, wie bereits beschrieben, genutzt werden.

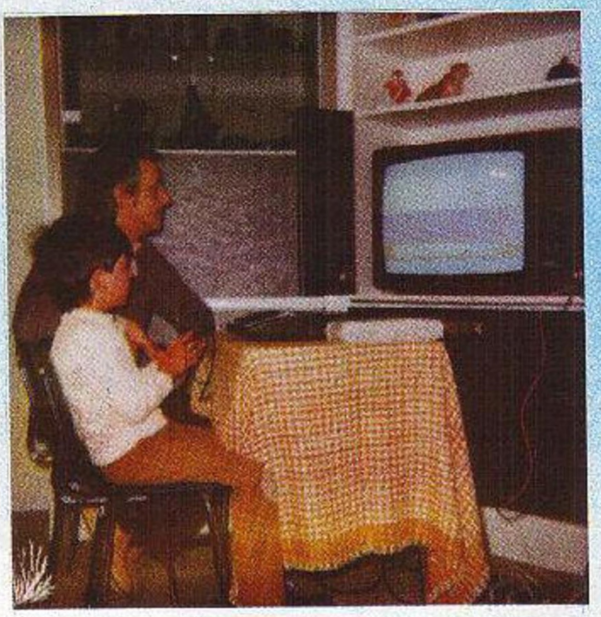
```
50 ! ****************************
51 ! *                          *
52 ! * JOERG HIRT               *
53 ! * BIRKENSTRASSE 10         *
54 ! * 6542 LIEBSHAUSEN         *
55 ! *                          *
56 ! ****************************
100 CALL CLEAR :: CALL DELSPRITE(ALL):: DISPLAY AT(3,1):"************************
*****" :: DISPLAY AT(4,1):"*                                  *"
110 DISPLAY AT(5,1):"*  C H A R    E D I T O R  *" :: DISPLAY AT(6,1):"*
                    *"
120 DISPLAY AT(7,1):"****************************" :: DISPLAY AT(13,4):"1 = STRI
NG ERSTELLEN" :: DISPLAY AT(17,4):"2 = STRING EINGEBEN"
130 CALL KEY(0,K,S):: IF S=0 THEN 130 :: IF K=49 THEN 140 ELSE IF K=50 THEN 680
ELSE IF K<49 OR K>50 THEN CALL SOUND(20,200,5):: GOTO 130
140 REM SPRITE EDITOR
150 IF GG=1 THEN 510
160 DIM B(16,16):: SC=1 :: C1=2 :: Z=1 :: S=1 :: HEX$="0123456789ABCDEF" :: M=1
:: GG=0
170 CALL CHAR(100,"0000000000000000FFFFFFFFFFFFFFFFFFFFC3C3C3C3FFFF")
180 CALL COLOR(9,2,16):: CALL CLEAR :: DISPLAY AT(1,10):"CHAR  EDITOR"
190 FOR Z=1 TO 16 :: CALL HCHAR(4+Z,2,100,16):: NEXT Z
200 CALL MAGNIFY(1):: IF GG=1 THEN GOTO 680 ELSE IF K=84 THEN GOTO 210 ELSE CALL
 SCREEN(8)
210 CALL DELSPRITE(ALL):: CALL SPRITE(#6,102,14,32,8):: CALL HCHAR(21,1,32,93)
220 DISPLAY AT(22,2):"E=UP,X=DOWN,S=LEFT,D=RIGHT" :: DISPLAY AT(23,2):"1 = PIXEL
  ON  0 = OFF" :: DISPLAY AT(24,2):"5 = MUSTERCODE AUSGEBEN"
230 CALL KEY(0,K,SS):: IF SS=0 THEN 230
240 IF K=48 THEN PIXEL=100 ELSE IF K=49 THEN PIXEL=101 ELSE IF K=53 THEN 310
250 IF K=83 THEN S=S-1 :: GOTO 290
260 IF K=68 THEN S=S+1 :: GOTO 290
270 IF K=69 THEN Z=Z-1 :: GOTO 290
280 IF K=88 THEN Z=Z+1 :: GOTO 290
290 IF S<1 THEN S=16 ELSE IF S>16 THEN S=1 ELSE IF Z<1 THEN Z=16 ELSE IF Z>16 TH
EN Z=1
300 CALL LOCATE(#6,(8*Z)+25,8*S+1):: CALL HCHAR(4+Z,1+S,PIXEL):: CALL SOUND(20,2
00,5):: GOTO 230
310 CALL DELSPRITE(ALL):: CALL HCHAR(21,1,32,128):: DISPLAY AT(23,2):"B I T T E
  W A R T E N"
```

```
320 FOR Z=1 TO 16 :: FOR S=1 TO 16 :: CALL GCHAR(4+Z,1+S,GC):: GC=GC-100 :: B(Z,
S)=GC :: NEXT S :: NEXT Z :: M$=""
330 FOR Z=1 TO 16
340 LOW=B(Z,5)*8+B(Z,6)*4+B(Z,7)*2+B(Z,8)+1
350 HIGH=B(Z,1)*8+B(Z,2)*4+B(Z,3)*2+B(Z,4)+1
360 M$=M$&SEG$(HEX$,HIGH,1)&SEG$(HEX$,LOW,1)
370 NEXT Z
380 FOR Z-1 TO 16
390 LOW=B(Z,13)*8+B(Z,14)*4+B(Z,15)*2+B(Z,16)+1

400 HIGH=B(Z,9)*8+B(Z,10)*4+B(Z,11)*2+B(Z,12)+1
410 M$=M$&SEG$(HEX$,HIGH,1)&SEG$(HEX$,LOW,1)
420 NEXT Z
430 DISPLAY AT(22,2):"C=COLOR  M=MAGNIFY  T=EDIT" :: DISPLAY AT(23,2):"A=ERASE,Q
=QUIT,B=BACKCOLOR"
440 DISPLAY AT(24,5):"L=LIST MUSTERCODE"
450 CALL CHAR(104,M$):: A=0
460 A=A+1 :: IF A=1 OR A=2 THEN CALL MAGNIFY(M):: CALL DELSPRITE(#5):: M=2
470 IF A=1 THEN CALL SPRITE(#1,104,C1,40,170,0,0,#3,106,C1,40,195,0,0,#2,105,C1,
70,170,0,0,#4,107,C1,70,195,0,0)
480 IF A=3 THEN M=3 :: CALL DELSPRITE(ALL):: CALL SPRITE(#5,104,C1,100,180,0,0)
490 IF A=3 OR A=4 THEN CALL MAGNIFY(M):: M=4
500 IF A=4 THEN A=0 :: M=1
510 CALL KEY(0,K,SS):: IF SS=0 THEN 510
520 IF K=67 THEN GOTO 550 ELSE IF K=77 THEN GOTO 460 ELSE IF K=84 THEN Z=1 :: GO
TO 200 ELSE IF K=65 THEN GOTO 140 ELSE IF K=81 THEN GG=0 :: GOTO 100
530 IF K=66 THEN GOTO 580 ELSE IF K=76 THEN 600
540 GOTO 510
550 C1=C1+1 :: IF C1>16 THEN 570 ::
560 CALL COLOR(#1,C1,#2,C1,#3,C1,#4,C1,#5,C1):: GOTO 510
570 C1=2 :: CALL COLOR(#1,C1,#2,C1,#3,C1,#4,C1,#5,C1):: GOTO 510
580 SC=SC+1 :: IF SC=17 THEN SC=2
590 CALL SCREEN(SC):: GOTO 510
600 CALL CLEAR :: CALL DELSPRITE(ALL):: DISPLAY AT(1,7):"PROGRAMM LISTING"
610 DISPLAY AT(5,1):"HEXADEZIMAL CODE :" :: DISPLAY AT(7,1):M$
620 M1$=SEG$(M$,1,16):: M3$=SEG$(M$,17,16):: M2$=SEG$(M$,33,16):: M4$=SEG$(M$,49
,16)
630 DISPLAY AT(11,1):"FELD 1 ";M1$ :: DISPLAY AT(12,1):"FELD 2 ";M2$ :: DISPLAY
AT(13,1):"FELD 3 ";M3$ :: DISPLAY AT(14,1):"FELD 4 ";M4$
640 DISPLAY AT(17,1):"SCREEN  ";SC :: DISPLAY AT(18,1):"COLOR   ";C1
650 DISPLAY AT(24,6):"D=DRUCKER,Q=QUIT"
660 CALL KEY(0,K,SS):: IF SS=0 THEN 650 :: IF K=68 THEN 670 ELSE IF K=81 THEN GG
=0 :: GOTO 100 ELSE IF K<>65 THEN GOTO 660
670 OPEN #1:"PIO" :: CALL CLEAR :: DISPLAY AT(15,4):"DRUCKER EINSCHALTEN" :: PRI
NT #1:"104 ";M$ :: CLOSE #1 :: GOTO 100
680 CALL CLEAR :: CALL DELSPRITE(ALL)
690 SC=1 :: C1=2 :: M$="" :: M1$="" :: M2$="" :: M3$="" :: M4$="" :: G1=0 :: G2=
0 :: G3=0 :: GG=0 :: M=1
700 DISPLAY AT(1,8):"STRING EINGEBEN" :: DISPLAY AT(5,1):"INPUT STRING ?:"
710 ACCEPT AT(8,1)SIZE(16)VALIDATE("0123456789ABCDEF")BEEP:M1$ :: ACCEPT AT(10,1
)SIZE(16)VALIDATE("0123456789ABCDEF")BEEP:M2$
720 ACCEPT AT(12,1)SIZE(16)VALIDATE("0123456789ABCDEF")BEEP:M3$ :: ACCEPT AT(14,
1)SIZE(16)VALIDATE("0123456789ABCDEF")BEEP:M4$
730 CALL GCHAR(10,3,G1):: CALL GCHAR(12,3,G2):: CALL GCHAR(14,3,G3)
740 IF G1=32 THEN M2$="0000000000000000"
750 IF G2=32 THEN M3$="0000000000000000"
760 IF G3=32 THEN M4$="0000000000000000"
770 M$=M1$&M2$&M3$&M4$
780 GG=1 :: GOTO 430
```

HOMECOMPUTER präsentiert:
Das erste *TOP-PROGRAMM* des Monats

Nanuk der Eskimo



Willi Döltsch und sein sechsjähriger Sohn Mike erstellten das erste Topprogramm für Homecomputer im Teamwork.

Tuut, tuut... - Hier Döltsch - Schönen guten Tag, Herr Döltsch. Hier ist der Roeske-Verlag in Eschwege, Redaktion HOMECOMPUTER. Sie haben uns das erste TOP-PROGRAMM des Monats geschickt: Herzlichen Glückwunsch!

Willi Döltsch aus Darmstadt war da denn doch ein wenig sprachlos, als wir vor wenigen Tagen bei ihm anklingelten, um ihn, den ersten glücklichen Gewinner unseres großen Dauerwettbewerbes um das beste Spielprogrammlisting, zu interviewen und unseren Lesern vorstellen zu können.

Ja, der kleine Eskimo NANUK, der sich so tapfer als Lebensretter seiner kleinen Pinguin-Freunde engagiert, hatte das Rennen gemacht und die Herzen der Jury im Sturm erobert!

Leicht hatten wir es uns von der Redaktion gewiß nicht vorgestellt, aus der Fülle der Einsendungen den Sieger zu ermitteln - und so kam es denn auch. So viele herrliche Programme brachte der Postbote uns in das Haus: mit phantastischer Graphik, tollen Sound-Effekten, Spannung, Abwechslung, einem guten Maß programmtechnischen Know-how und immer wieder neuen Ausgangssituationen.

Da hatten wir fast ein schlechtes Gewissen, uns für einen einzigen Sieger entscheiden zu müssen. Ein freundschaftliches Dankeschön an alle Computer-Fans, die mitgemacht und uns ihr Top-Programm zugesandt haben!!

Doch nun zu unserem Spitzen-Programmierer:

Willi Döltsch ist 37 Jahre alt, seit 10 Jahren verheiratet und hat einen sechsjährigen Sohn, dem es auch schon gehörig in den Fingern juckt, wenn er vor Tastatur und Joystick sitzt.

Unser Leser mit dem Rettungsprogramm für Pinguine ist von Beruf Leiter der technischen EDV-Abteilung eines großen und bekannten Bekleidungsherstellers und vornehmlich mit der Erstellung, Organisation und Optimierung von CAD-Software beschäftigt. Wie viele unserer begeisterten Hobby-Programmierer wissen werden, stehen diese drei Buchstaben für "Computer Aided Design" und meinen eine "rechnerunterstützte Formgestaltung".

Mancher wird ein solches System im Automobilbau oder großen Architekturbüros, nicht aber in den Fertigungsstätten eines Kleiderfabrikanten vermuten.

Wenn er aber bedenkt, daß gute Paßform bei nicht individuell gefertigter Konfektionsware oft zum reinen Glücksfall werden kann und bei nicht optimalem Zuschnitt viel wertvolles Material verloren geht, wird er ermessen können, daß ein verantwortungsvoller Unternehmer auch die Investition in eine fast 2 Millionen Mark teure Rechenanlage nicht zu scheuen braucht. Wir fragten Willi Döltsch natürlich, ob ein EDV-Profi nicht froh sei, unter Umständen am Feierabend bzw. Wochenende mal ohne den Computer, der doch den Arbeitsalltag prägt, auskommen zu können; zu verargen wäre es ihm bestimmt nicht. - Herr Döltsch winkt entschieden ab! Ganz und gar nicht, meint er.

Gerade als ein Mann, der es ständig mit einer leistungsfähigen Hardware zu tun hat, die unweigerlich auch den Spieltrieb anregt, jedoch nur für ernste technische Belange genutzt werden darf, freut er sich wie ein Schneekönig (oder sollte man in unserem Falle Eskimo sagen?) auf seinen kleinen Texas TI-99, der ihm in BASIC das bieten kann, wozu er am Arbeitsplatz in Fortran-IV keine Gelegenheit hat, nämlich seiner spielerischen Fantasie die Zügel schießen zu lassen...

Gut, wenden wir ein, als Junggeselle mag man in solch eigenbrödlerischer Weise über seine Freizeit verfügen können; der Familienvater sollte da aber wohl auf massiven Widerstand seiner besseren Hälfte stoßen. In dieser Hinsicht kann ich wirklich nicht klagen, meint HOMECOMPUTER-Leser Döltsch. Meine Frau ist da ein echter Schatz und hat eine Menge Verständnis für mich und meine (zweite) Leidenschaft, obgleich sie durch das Thema ROM und RAM nicht in dieselbe Euphorie gerät wie ihr Ehegesponst. Dazu kommt aber noch, daß auch sie ein Steckenpferd reitet, das viel Zuwendung erfordert - sie hält einen Graupapagei (der vielleicht irgendwann auch in einem Computerspiel eine Wiedergeburt erleben mag) und Hand auf's Herz: wenn die Ehefrau einen Vogel hat, kann sie ihrem Gemahl den TI-99 nicht verbieten, oder? Wir widersprachen nicht!

Um nun auf die Ursache für das Interview zu sprechen zu kommen, erzählt uns Herr Döltsch, daß es Filius Mike gewesen sei, der die Idee mit dem kleinen Eskimo hatte. Und weil ein Sechsjähriger durchaus die künstlerische Freiheit verantworten darf, eine antarktische Tierart in die Heimat der Eskimos im nordischen Packeis geraten zu lassen, wurde der Plan von Vater Willi alsbald in Angriff genommen - für Eskimo Nanuk sollte das Ganze aber zum Horrortrip werden....

Irgendwo am Polarkreis ist eine kleine Pinguin-Kolonie durch vier Eisbären auf's Ärgste bedroht. Ohne Nanuk ist sie dem sicheren Tode verfallen. Nun befindet sich zwischen dem Eskimo auf dem Festland und den armen Vögeln auf dem schwimmenden Eisberg eine tüchtige Wegstrecke Eismeer, die ein ernstes Hindernis darstellt. Hilfe leistet nur eine Walherde, die in den Fluten herumplantscht: Über die Rücken der Meeressäuger hinweg kann Nanuk zu den Pinguinen gelangen. Unser Nanuk ist ein wahrer Held, der über sage und schreibe fünf Leben verfügt. Dies hindert ihn aber nicht daran, sich (wie alle Eskimos) zu den passionierten Nichtschwimmern zu rechnen. Dieser Umstand relativiert die Lebenstüchtigkeit Nanuks natürlich gewaltig, da auch der kühnste Nordmann bei seinen Sprüngen von Wal zu Wal ins Eismeer plumpsen und kläglich auf tödliche Tauchstation gehen kann. Kurzum, eine teuflische Situation, in der Nanuk da steckt, denn auf dem Eisberg warten ja die Bären!

Das Spiel ist sehr gut aufgebaut, mit witzigen Details ausgestattet und ohne streckende langweilige Passagen - ein Spiel, wie wir Homecomputer-Freaks es wünschen. Es wird allen unseren Lesern wohl ebenso viel Spaß bereiten wie uns beim Testlauf.

Willi Döltsch ist eine Einladung nach Eschwege bereits sicher. Er würde sich freuen, dort mit den nächsten Gewinnern zusammenkommen zu dürfen, um Erfahrungen auszutauschen und sich die Räumlichkeiten anschauen zu können, wo HOMECOMPUTER und CPU entstehen. "Seine" Zeitschrift HOMECOMPUTER sammelt er übrigens seit der ersten Nummer und baut sich eine preiswerte Programm-Bibliothek auf. Ich kam auf HOMECOMPUTER, so sagt er, weil es das meiner Ansicht nach erste Fachmagazin gewesen ist, das schöne und anspruchsvolle Software in puncto Spiel für "meinen" TI-99 herausbrachte. Ein Lob nicht so sehr an uns als vielmehr unsere Leser, von denen die meisten Programme stammen - und ein Lob aus kompetentem Mund.

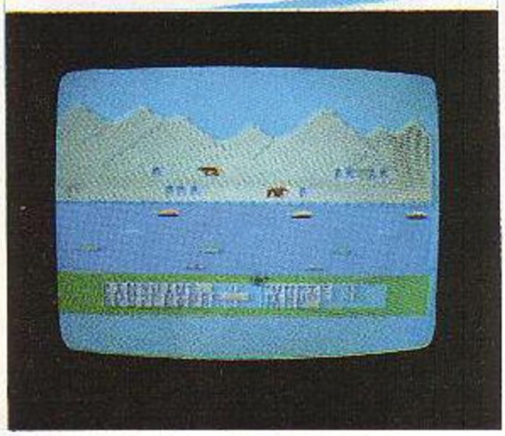
Wer macht es ihm nach und schreibt unser nächstes TOP-PROGRAMM?

TOPPROGRAMM

für den TI-99/4A

Nanuk der Eskimo

für den TI-99/4A



Im nordischen Packeis bangt eine Pinguinherde um ihr Leben. Bedroht ist es durch eine Gruppe von vier Eis- und Braunbären, die leider vegetarische Kost verabscheuen. Jenseits eines Eismeerabschnittes befindet sich ein möglicher Retter, nämlich Nanuk, unser Held.

Nanuk kann, da er Nichtschwimmer ist, lediglich mit Hilfe einer Walfamilie zu seinen Schützlingen gelangen. Die Wale schwimmen, Rücken über Wasser, fröhlich umher und steigern im Verlauf des Spieles ihre Geschwindigkeit.

Nanuk besitzt fünf Leben, die er dadurch einbüßt, daß er entweder von einem Walrücken abrutscht und im Wasser ersäuft oder aber von den Eisbären verspeist wird.

Steuerbar ist Nanuk mit dem Joystick in alle vier Richtungen. Retten kann er einen Pinguin, indem er sich vor diesen stellt und es dem gütigen Spieler anheim stellt, mit dem Joystick zu feuern.

Tragisch wird die Situation für Nanuk und den Joystick-Kommandanten, wenn unser Eskimo während einer Rettungsaktion verunglückt. Dann ist nicht nur eines seiner eigenen Leben über den Jordan gegangen, sondern auch der ihm jeweils anvertraute Pinguin zu beklagen.

Ziel des Spieles ist es, mit möglichst wenig Verlusten an Pinguinen und Eskimoleben über die Runden zu kommen. Das ist nicht einfach...

ANZEIGE: Jeweils höchster Punktestand, Name des besten Spielers (max. sechs Buchstaben ohne X,Y,Z).

SPIELENDE: Alle Pinguine gerettet oder alle Pinguinleben erloschen.

Noch ein paar wichtige Hinweise zum Eintippen des Programmes:

Manche Befehlszeilen werden wegen ihrer Länge nicht ganz angenommen. Beenden Sie die Eingabe mit > ENTER< , und rufen Sie dann diese Befehlszeile durch gleichzeitiges Drücken von > FCTN und REDO< zurück auf den Bildschirm. Jetzt können Sie die fehlenden Befehle fertig schreiben.

In diesem Programm werden frei definierte Zeichen benutzt, die mit > DISPLAY AT< plaziert werden. Diese Zeichen werden im Listing normal nicht ausgedruckt. Zur besseren Eingabe sind für diese Zeichen Buchstaben und Zeichen ausgedruckt, die bei "gedrückter CTRL-Taste" eingegeben werden müssen. Es erscheinen dabei keine Zeichen!!

Das ist kein Fehler!! Erst wenn das Programm mit > RUN< gestartet wird, sind diese Zeichen beim Listen des Programmes sichtbar.

Folgende Programmzeilen sind auf diese Weise einzugeben:

Zeilennummer 450 bis 510.

Zum Spielen muß die ALPHA-LOCK Taste gelöst (oben) sein.

Wenn der Bestspieler seinen Namen eingeben kann, sind alle Buchstaben bis auf > X< > Y< > Z> erlaubt.

Die "@"-Zeichen im Listing sind miteinzugeben.

Zur Programmiertechnik:

Ein großes Problem war die Programmierung der Bewegungsabläufe der vier Bären sowie die Sprungtechnik von "NANUK".

Durch die "SPRITE-TECHNIK" des Extended Basic ist es möglich, eine relative Bewegung (mit CALL MOTION) sowie eine absolute Bewegung (mit CALL LOCATE) zu erzeugen. Beide Techniken mußten hier aufeinander abgestimmt werden.

Mit Sprites zu arbeiten, bringt immer das Problem mit sich, wie man am sichersten ein Zusammentreffen von 2 oder mehreren Sprites über das Programm kontrollieren kann. In diesem Fall war es das Zusammentreffen des Eskimos mit einem Wahlfisch bzw. einem Bären.

Da die Pinguine nicht als Sprites aufgebaut wurden, mußte hier ein Kontakt zwischen dem Eskimo und einem Pinguin über den "CALL GCHAR" (=CALL GET CHARACTER) Befehl erzeugt werden.

Mein Ziel war es, über den ganzen Spielablauf hin, keinen Bewegungsstillstand auf dem Bildschirm zu haben.

W. Döltsch

```
100 CALL CLEAR :: CALL SCREEN(8):: CALL MAGNIFY(3):: CALL COLOR(14,16,8,13,15,16
,8,3,3,2,5,16):: RANDOMIZE :: DIM M1$(25),M2$(25)
110 FOR I=3 TO 7 :: CALL COLOR(I,5,16):: NEXT I
120 M=200 :: Z=145 :: S=128 :: F0=116 :: F0=1 :: SD=5 :: LIV=5 :: PR=10 :: PIN=0
130 DATA 1,2,139,2,1,137,1,31,141,2,32,142
140 DATA TOLLE@SACHE@,SABENHAFT@@@,PHANTASTISCH,GUT@GEMACHT@,BESTENS@@@@@,TOP@@GE
MACHT@,SEHR@GUT@@@@@,HERVORAGENDE@
150 DATA BRAVISSIMO@@@,GRATULATION@,WAHNSINN@@@@@,GUT,BLEIFUSS,NA@JA.....@@,KOENNE
RHAFT@,DU@SUPERSTAR,ACH@WIE@GUT@
160 DATA HUT@AB...@@@,GUT,GUT,GUT..SCHON@BESSER,ALLES@KLAR@@,NA@EITTE@@@@@,NOCH@E
INMAL@,BUSSI@BUSSI@,VIELEN@DANK@,NA@SOWAS....
170 RESTORE 140 :: FOR I=1 TO 25 :: READ M1$(I):: NEXT I
180 DATA OBERPFLAUME@,SUPERNIETE@@@,WASSERKOPF@@,KAULQUAPPE...BLUT@FLIESST,BLINDG
AENGER,HILFE,HILFE..ARMER@IRRER@,GESTORREN..
190 DATA GRASHUEPFER@,AUFPASSEN....OBERFLASCHE@,TAUBC@NUSS@@,AUGEN@AUF@@@,SAUSCH
LECHT@,OBERFAUL@@@@@,STARFIGHTER@,BANTU@NEGER@
200 DATA WOYL@BLIND...TOLLPATSCH@@@,FLEDERMAUS@@@,OH@NEIN.....@,MAMA@MIA....@,D@
WEIA@C@WEI@,WAS@SOLL@DAS@
210 RESTORE 190 :: FOR I=1 TO 25 :: READ M2$(I):: NEXT I
220 CALL CHAR(47,RPT$("F",16),64,RPT$("0",16))
230 CALL CHAR(132,"0102020C08107080",133,"8040483402020101",134,"0304040838408 08
0",135,"8080403010 0C0201")
240 CALL CHAR(136,"0001010 70F1F7FFF",137,"0103032F7F7FFFFF",138,"FFFFFFFFFFFFCFC
6",139,"0307071F1FDFFFFF")
250 CALL CHAR(140,"C0C0E4FEFFFEFFFF",141,"80D0F8FCFEFEFFFF",142,"80C0E0F8FCFCFEF
F",143,"FFFFFFFFFFFFFFFF")
260 W$(1)="0000000001187DFFF"&RPT$("0",24)&"F8FEFBFF"&RPT$("0",16)
270 W$(2)="0000000001F7FDFFF"&RPT$("0",24)&"80E1FBFF"&RPT$("0",16)
280 P$(1)="1828781824561866" :: P$(2)="1868186624182466"
290 BL$(1)=RPT$("0",13)&"8182F7F0F0303030F0C"&RPT$("0",16)&"F8FCFCFE8F870307"
300 BL$(2)=RPT$("0",15)&"81F377F030307C103"&RPT$("0",16)&"F8FCFEFE8E9E868E"
310 BL$(3)=RPT$("0",16)&"171F2F7B03070001"&RPT$("0",12)&"1018F4FEEC0C0C0C0E070"
320 BR$(1)=RPT$("0",16)&"1F3F7F7FF1E1C0E0"&RPT$("0",14)&"10F8F4FEC0F00080"
330 BR$(2)=RPT$("0",16)&"1F3F7F7F71796171"&RPT$("0",16)&"E8F8F4DEC0F00080"
340 BR$(3)=RPT$("0",16)&"1F3F7F7F737B333B"&RPT$("0",16)&"E8F8F4DEC0F00080"
350 E$(1)="0000000002070A05020F12171505050F0000000000000080001C8C5460408000E0"
360 E$(2)="0000020700A05221F03024D70000000000000001C0C9420406000000907000000000"
370 E$(3)="0000020700A05221F03024D70000000000000001C0C9420406000000907000000000"
380 E$(4)=RPT$("0",19)&"1031512090701"&RPT$("0",20)&"8050902 0C000"
390 E$(5)=RPT$("0",19)&"103050339C701"&RPT$("0",20)&"80408038C000"
400 CALL CHAR(36,W$(1),40,W$(2))
410 CALL CHAR(92,BL$(1),96,BL$(2),100,BL$(3))
420 CALL CHAR(104,BR$(1),108,BR$(2),112,BR$(3))
430 CALL CHAR(116,E$(1),120,E$(2),124,E$(3),128,E$(4),60,E$(5))
440 CALL CHAR(45,P$(1))
450 DISPLAY AT(1,1):"L----HL----------KN--KL----I"
460 DISPLAY AT(2,1):"OMIMKOJN--IL--KMKOOLKOOM--IO"
470 DISPLAY AT(3,1):"OFOJGCOOLHCOMKOFJOCFOOOONHOO"
480 DISPLAY AT(4,1):"DOOOOOOOJGOUDOOOOOFOOOOOOGOO"
490 DISPLAY AT(5,1):"DOOOOUOCOOEOOOOOOFOOCOOOOOEO"
500 DISPLAY AT(6,1):"DOOOOJOOOOOOOOOOOFOCUOOOOOOOO"
510 DISPLAY AT(7,1):"DOOOCOOOOOOOOOOOFOCUOOOOOOOO"
520 FOR I=1 TO 31 STEP 30 :: CALL VCHAR(2,I,143,9):: NEXT I :: FOR I=2 TO 32 STE
P 30 :: CALL VCHAR(2,I,143,9):: NEXT I
530 RESTORE 130 :: FOR I=1 TO 4 :: READ A,B,C :: CALL HCHAR(A,B,C):: NEXT I
540 CALL HCHAR(8,1,143,96):: CALL HCHAR(11,1,138,32):: CALL HCHAR(12,1,47,256)::
 CALL HCHAR(20,1,92,160)
550 GOSUB 1380 :: DISPLAY AT(21,5)SIZE(16):"NANUK,DER@ESKIMO"
560 DISPLAY AT(22,7)SIZE(14):"@@ALI@RIGHTS@@" :: DISPLAY AT(23,5)SIZE(18):"@WILL
I@H,DOELTSCH@"
570 GOSUB 1200
580 CALL SPRITE(#2,92,1,48,256,0,-12,#3,104,1,65,1,0,4,#4,92,1,48,100,0,-12)
590 CALL SPRITE(#5,104,1,65,128,0,4)
600 CALL SPRITE(#6,36,1,91,1,0,SD,#7,36,1,91,1[*8,0,SD,#8,36,1,91,22*8,0,SD)
610 CALL SPRITE(#9,40,1,107,3*8,0,-SD,#10,40,1,107,13*8,0,-SD,#11,40,1,107,24*8,
0,-SD)
620 CALL SPRITE(#12,36,1,123,5*8,0,SD,#13,36,1,123,15*8,0,SD,#14,36,1,123,26*8,0
,SD)
630 CALL SPRITE(#15,40,1,139,7*8,0,-SD,#16,40,1,139,17*8,0,-SD,#17,40,1,139,28*8
,0,-SD)
640 GOSUB 1380
650 DISPLAY AT(21,3)SIZE(8):"PINGUINE" :: DISPLAY AT(23,3)SIZE(9):"RESTLEBEN" ::
 DISPLAY AT(21,16)SIZE(6):"PUNKTE" :: DISPLAY AT(22,16)SIZE(6):"HIGHSC"
660 DISPLAY AT(23,16)SIZE(4):"NAME"
```

```
670 FOR I=21 TO 23 :: CALL HCHAR(I,15,32,3):: NEXT I :: FOR I=21 TO 22 :: CALL H
CHAR(I,24,32,5):: NEXT I :: CALL HCHAR(23,23,32,6)
680 DISPLAY AT(21,12)SIZE(3):PIN :: DISPLAY AT(23,12)SIZE(3):LIV :: DISPLAY AT(2
1,22)SIZE(5):PUN :: DISPLAY AT(22,22)SIZE(5):HIC
690 DISPLAY AT(23,21)SIZE(6):N$ :: DISPLAY AT(22,3)SIZE(12):"KOMMENTARE@@"
700 CALL COLOR(#2,7,#3,7,#4,15,#5,15,#6,10,#7,8,#8,4,#9,15,#10,4,#11,15)
710 CALL COLOR(#12,10,#13,8,#14,8,#15,4,#16,15,#17,10)
720 CALL SPRITE(#1,116,2,Z,S):: CALL SOUND(-300,1800,0):: V=123
730 CALL MOTION(#1,0,0):: CALL SOUND(5,-6,1):: :: IF FLA=1 THEN GOSUB 1260 :: CA
LL SOUND(-300,1800,0)
740 IF FLA2=1 THEN GOTO 770 ELSE FLA2=1 :: CALL POSITION(#1,Z,S):: S=INT(S/8+.9)
*8 :: V=S/8
750 IF V=1 THEN CALL LOCATE(#1,Z,1)ELSE CALL LOCATE(#1,Z,S):: GOTO 770
760 IF V=32 THEN CALL LOCATE(#1,Z,256)ELSE CALL LOCATE(#1,Z,S)
770 CALL JOYST(1,X,Y):: IF X=0 AND Y=0 THEN GOSUB 1160 :: GOTO 770
780 IF Y=4 THEN 790 :: IF Y=-4 THEN 830 :: IF X=4 THEN 850 :: IF X=-4 THEN 880 E
LSE 770
790 PO=PO+1 :: FLA2=0 :: GOSUB 1180 :: Z=Z-8 :: CALL LOCATE(#1,Z,S)
800 CALL PATTERN(#1,120):: Z=Z-4 :: CALL LOCATE(#1,Z,S)
810 CALL PATTERN(#1,124):: GOSUB 1160 :: Z=Z-4 :: CALL LOCATE(#1,Z,S)
820 CALL PATTERN(#1,116):: GOSUB 1160 :: GOTO 1150
830 PO=PO-1 :: IF PO=0 THEN PO=1 :: GOTO 770 ELSE GOSUB 1180 :: Z=Z+8 :: CALL LO
CATE(#1,Z,S):: CALL PATTERN(#1,120):: GOSUB 1160 :: Z=Z+4 :: CALL LOCATE(#1,Z,S)
840 CALL PATTERN(#1,124):: Z=Z+4 :: CALL LOCATE(#1,Z,S):: CALL PATTERN(#1,116)::
 GOSUB 1160 :: GOTO 1150
850 V=S+8 :: IF V>248 THEN CALL LOCATE(#1,Z,1):: S=1
860 IF PO>1 AND PO<6 THEN 1060 ELSE S=S+2 :: CALL LOCATE(#1,Z,S):: CALL PATTERN(
#1,120):: GOSUB 1160 :: S=S+2 :: CALL LOCATE(#1,Z,S)
870 CALL PATTERN(#1,124):: S=S+4 :: CALL LOCATE(#1,Z,S):: CALL PATTERN(#1,116)::
 GOTO 1150
880 V=S-8 :: IF V<8 THEN CALL LOCATE(#1,Z,256):: S=256
890 IF PO>1 AND PO<6 THEN 1060 ELSE S=S-2 :: CALL LOCATE(#1,Z,S):: CALL PATTERN(
#1,120):: GOSUB 1160 :: S=S-2 :: CALL LOCATE(#1,Z,S)
900 CALL PATTERN(#1,124):: S=S-4 :: CALL LOCATE(#1,Z,S):: CALL PATTERN(#1,116)::
 GOTO 1150
910 FOR I=1 TO 2 :: CALL COINC(ALL,HIT):: IF HIT THEN CALL MOTION(#1,0,-SD):: CA
LL SOUND(-50,-1,0):: GOTO 770
920 NEXT I
930 GOTO 1060
940 FLA1=0 :: FOR I=1 TO 2 :: CALL COINC(ALL,HIT):: IF HIT THEN CALL MOTION(#1,0
,SD):: CALL SOUND(-50,-2,0):: GOTO 770
950 NEXT I
960 GOTO 1060
970 CALL MOTION(#1,0,0):: CALL COINC(ALL,HIT):: IF HIT THEN 1190
980 IF FLA1=1 THEN GOTO 1010 ELSE FLA1=1 :: CALL POSITION(#1,Z,S):: S=INT(S/8+.9
)*8
990 IF S<8 THEN CALL LOCATE(#1,Z,1):: S=1 ELSE CALL LOCATE(#1,Z,S):: GOTO 1010
1000 IF S>248 THEN CALL LOCATE(#1,Z,256):: S=256 ELSE CALL LOCATE(#1,Z,S)
1010 FOR I=1 TO 2 :: CALL COINC(ALL,HIT):: IF HIT THEN 1190
1020 NEXT I :: GOSUB 1160
1030 IF FLA=1 THEN GOTO 1050 ELSE CALL KEY(1,T,ST):: IF T<>18 THEN GOTO 1050 ELS
E CALL GCHAR(INT(Z/8+2),S/8+1,C):: IF C<>45 THEN GOTO 1050 ELSE FLA=1
1040 CALL SOUND(1,2500,0):: CALL HCHAR(INT(Z/8+2),INT(S/8+1),143):: CALL HCHAR(2
1,3,45)
1050 CALL JOYST(1,X,Y):: IF X=0 AND Y=0 THEN 1010 ELSE 780
1060 CALL MOTION(#1,0,0):: CALL PATTERN(#1,128):: CALL HCHAR(21,3,94):: CALL SOU
ND(200,-5,5):: GOSUB 1160
1070 FOR I=1 TO 13 :: CALL PATTERN(#1,60):: GOSUB 1160 :: CALL PATTERN(#1,128)::
 GOSUB 1160 :: NEXT I
1080 CALL DELSPRITE(#1):: Z=144 :: S=128 :: IF FLA=0 THEN 1110
1090 GOSUB 1160 :: PUN=PUN-250 :: CALL HCHAR(21,3,94):: IF PUN<0 THEN PUN=0
1100 DISPLAY AT(21,22)SIZE(5):PUN :: PR=PR-1 :: IF PR=0 THEN GOTO 1460
1110 GOSUB 1160 :: PO=1 :: FLA=0 :: LIV=LIV-1 :: IF LIV=1 THEN GOSUB 1420
1120 DISPLAY AT(23,12)SIZE(3):LIV :: DISPLAY AT(22,3)SIZE(12):M2$(INT(RND*25)+1)
1130 IF LIV=0 THEN GOTO 1460 ELSE GOSUB 1160 :: GOTO 720
1140 CALL PATTERN(#1,120):: CALL MOTION(#1,19,0):: FOR I=1 TO 7 :: GOSUB 1160 ::
 NEXT I :: GOTO 1060
1150 ON PO GOTO 730,910,940,910,940,970,1010,1140
1160 CALL PATTERN(#2,FO-24,#3,FO-12,#4,FO-24,#5,FO-12):: FO=FO+4 :: IF FO=128 TH
EN FO=116
1170 RETURN
1180 CALL POSITION(#1,Z,S):: RETURN
1190 FOR I=113 TO 110 STEP -1 :: CALL SOUND(-20,1,5):: NEXT I :: GOTO 1080
1200 FOR I=1 TO 5
```

```
1210 SP=INT(RND*(29-4+1))+4
1220 CALL GCHAR(8,SP,C):: IF C=45 THEN 1210 ELSE CALL HCHAR(8,SP,45)
1230 SP=INT(RND*(30-3+1))+3
1240 CALL GCHAR(10,SP,C):: IF C=45 THEN 1230 ELSE CALL HCHAR(10,SP,45)
1250 NEXT I :: RETURN
1260 FLA=0 :: PIN=PIN+1 :: SD=SD+1 :: EX=INT(RND*(350-100+1))+100 :: IF EX>180 T
HEN CALL SOUND(150,783,5,987,5,174,5)
1270 PUN=PUN+550+EX :: DISPLAY AT(21,22)SIZE(5):PUN :: GOSUB 1160
1280 CALL HCHAR(21,3,94):: DISPLAY AT(21,12)SIZE(3):PIN :: DISPLAY AT(22,3)SIZE(
12):M1$(INT(RND*25)+1):: GOSUB 1160 :: IF PIN=10 THEN GOTO 1440
1290 PR=PR-1 :: IF PR=0 THEN GOTO 1460
1300 FOR I=1 TO 7 :: CALL CHAR(45,P$(2)):: CALL SOUND(-100,2700,5)
1310 GOSUB 1160 :: CALL CHAR(45,F$(1)):: CALL SOUND(-100,3300,5):: GOSUB 1160 ::
NEXT I
1320 FOR I=6 TO 14 :: IF I<9 OR I>11 THEN CALL MOTION(#I,0,SD)
1330 NEXT I :: GOSUB 1160
1340 FOR I=9 TO 17 :: IF I<12 OR I>14 THEN CALL MOTION(#I,0,-SD)
1350 NEXT I :: GOSUB 1160
1360 RETURN
1370 LIV=5 :: PR=10 :: I=145 :: S=120 :: PUN,EX,PIN,FLA,NN=0 :: SD=4 :: FO=116 :
: PO=1 :: GOSUB 1410 :: GOTO 550
1380 FOR I=21 TO 23 :: CALL HCHAR(I,5,64,24):: NEXT I :: RETURN
1390 DISPLAY AT(8,3)SIZE(23):"BESTSPIELER.IHR@NAME..." :: DISPLAY AT(9,3)SIZE(16
):"BIS@6@BUCHSTABEN."
1400 ACCEPT AT(9,20)SIZE(6)VALIDATE(UALPHA):N$ :: IF N$="" THEN N$="@@W.D."
1410 FOR I=8 TO 10 :: CALL HCHAR(I,3,143,26):: NEXT I :: RETURN
1420 FOR I=1 TO 4 :: DISPLAY AT(19,3)SIZE(24):"ACHTUNG.@NOCH@@1@ESKINO@" :: GOSU
B 1160 :: CALL SOUND(100,600,10,700,10,-2,10):: GOSUB 1160
1430 CALL HCHAR(19,5,47,24):: GOSUB 1160 :: NEXT I :: RETURN
1440 T=250 :: CALL SOUND(T,130,3,164,3):: GOSUB 1160 :: CALL SOUND(T,164,2,195,2
):: GOSUB 1160 :: CALL SOUND(T,195,1,261,2)
1450 GOSUB 1160 :: CALL SOUND(T*4,261,1,329,1,391,1):: FOR I=1 TO 5 :: GOSUB 116
0 :: NEXT I
1460 FOR I=6 TO 11 :: CALL DELSPRITE(#I):: NEXT I :: GOSUB 1160
1470 DISPLAY AT(13,3)SIZE(24):".......SPIELENDE....." :: DISPLAY AT(14,3)SIZE
(24):"@ABER@EINE@GUTE@LEISTUNG"
1480 DISPLAY AT(15,3)SIZE(24):"@@@@@FUER@DEN@ANFANG.@@@" :: GOSUB 1160 :: GOSUB
1590
1490 IF PUN<=HIC THEN 1520 ELSE HIC=PUN :: NN=1
1500 DISPLAY AT(13,3)SIZE(24):"@@@SIE@SIND@DER@BESTE.@@" :: DISPLAY AT(14,3)SIZE
(24):"@EINFACH@KLASSE@GEMACHT."
1510 DISPLAY AT(15,3)SIZE(24):"@@MACHEN@SIE@WEITER@SO.@" :: GOSUB 1160 :: GOSUB
1590
1520 DISPLAY AT(13,3)SIZE(24):"WOLLEN@SIE@WEITERSPIELEN" :: DISPLAY AT(14,3)SIZE
(24):"UND@MEHR@PUNKTE@SAMMELN."
1530 DISPLAY AT(15,3)SIZE(24):"DRUECKE@@@@.J.@ODER@.N.@" :: GOSUB 1160
1540 FOR D=1 TO 10 :: NEXT D :: CALL KEY(3,T,S):: IF S=0 THEN GOSUB 1160 :: GOTO
1540
1550 IF T<>74 THEN GOTO 1580
1560 FOR I=13 TO 15 :: CALL HCHAR(I,5,47,23):: NEXT I :: CALL DELSPRITE(ALL):: I
F NN=1 THEN GOSUB 1390
1570 GOTO 1370
1580 CALL CLEAR :: END
1590 GOSUB 1650
1600 CALL SOUND(M,415,3):: GOSUB 1160 :: CALL SOUND(M,494,3):: GOSUB 1160 :: CAL
L SOUND(M,523,2,110,15):: GOSUB 1160
1610 CALL SOUND(M,165,4):: GOSUB 1160 :: CALL SOUND(M,220,4):: GOSUB 1160 :: CAL
L SOUND(M,330,3):: GOSUB 1160 :: GOSUB 1650
1620 CALL SOUND(M,523,3):: GOSUB 1160 :: CALL SOUND(M,494,4):: GOSUB 1160 :: CAL
L SOUND(M,440,4,110,14):: GOSUB 1160 :: CALL SOUND(M,165,10)
1630 CALL SOUND(M,220,7):: GOSUB 1160 :: GOTO 1490
1640 CALL SOUND(M,330,6):: GOSUB 1160 :: CALL SOUND(M*3,440,6):: GOSUB 1160 :: G
OTO 1590
1650 CALL SOUND(M,659,6):: GOSUB 1160 :: CALL SOUND(M,622,6):: GOSUB 1160 :: CAL
L SOUND(M,659,6):: GOSUB 1160
1660 CALL SOUND(M,622,5):: GOSUB 1160 :: CALL SOUND(M,659,4):: GOSUB 1160 :: CAL
L SOUND(M,494,3):: GOSUB 1160 :: CALL SOUND(M,387,4):: GOSUB 1160
1670 CALL SOUND(M,523,5):: GOSUB 1160 :: CALL SOUND(M,440,6,110,15):: GOSUB 1160
1680 CALL SOUND(M,165,8):: GOSUB 1160 :: CALL SOUND(M,220,6):: GOSUB 1160 :: CAL
L SOUND(M,262,4):: GOSUB 1160 :: CALL SOUND(M,330,4):: GOSUB 1160
1690 CALL SOUND(M,440,4):: GOSUB 1160 :: CALL SOUND(M,494,4,131,15):: GOSUB 1160
1700 CALL SOUND(M,165,4):: GOSUB 1160 :: CALL SOUND(M,208,4):: GOSUB 1160 :: CAL
L SOUND(M,330,4):: GOSUB 1160 :: RETURN
1710 DISPLAY AT(5,10)SIZE(4):S :: RETURN
```

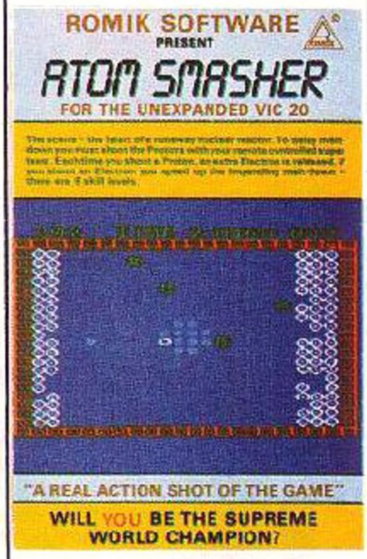
REVIEWS

Scuba Dive für den Commodore 64, Oric-1 und Spectrum 48K

Als Handlungsort für Computerspiele werden gern fantastische und geheimnisvolle Schauplätze gewählt. Die meisten bevorzugen den Weltraum. Einige - und so auch das hier vorgestellte - den Meeresgrund.
Die ausgezeichnete, an Comikstrips erinnernde Grafik und die schnell und ruckfrei ablaufenden Bewegungen weisen darauf hin, daß dieses Programm bei allen drei getesteten Versionen in Maschinensprache programmiert ist.
Der Spieler kontrolliert einen Taucher, der auf dem Meeresgrund zwischen Haifischen, Kraken und anderen Meeresbewohnern umherschwimmt. Seine Aufgabe ist es, den Eingang einer Unterwasserhöhle auszumachen. Durch das Labyrinth dieser Höhle muß er sich hindurchkämpfen, um an den sagenhaften Perlenschatz zu gelangen. Einige der Gänge sind Sackgassen, in anderen lauern grauenhafte Monster. Der Taucher hat natürlich nicht unbegrenzt Zeit für seine Suchaktion, da ihm nur ein bestimmter Sauerstoffvorrat zur Verfügung steht. Gelingt es ihm nicht rechtzeitig aufzutauchen, bedeutet dies sein Ende.
Sicher erinnert diese Geschichte an dutzende ähnlich gearteter Spiele. Durch die ausgezeichnete Programmierung stellt dieses Programm von Durrell-Software aus England jedoch einen besonderen Leckerbissen für die Freunde von Arcadespielen dar.

Atom Smasher für den VC-20 o. Erw.

Unseren Lesern, die in die Spuren von Otto Hahn, Ilse Meitner und Fritz Straßmann treten wollen, sei dieses Spiel wärmstens empfohlen. Von Romik aus England haben wir schon manches hübsche Spiel bekommen - was sie uns diesmal über den Kanal geschickt haben, halten wir jedoch für besonders originell.
Statt uns in galaktischen Dimensionen herumzutreiben, begeben wir uns diesmal in die Welt des Kleinsten: Ihre Aufgabe besteht darin, mit Hilfe eines Protons den Atomkern zu treffen, ohne dabei mit den Kern umkreisenden Elektronen Kontakt aufzunehmen. Jede Kollision zwischen Ihrem Proton und einem Elektron führt (in Abwechslung zu ähnlichen Spielen) zu einem Elektron mehr, wodurch sich Ihre Arbeit mit der Zeit erheblich erschweren kann. Dazu kommt, daß Ihr Proton eine Halbwertszeit von 1,5 besitzt, nach Einsatz von drei "Leben" somit kampfunfähig wird.
Programmtechnisch ist Atom Smasher zwar nicht in die Spitzenkategorie der Romik-Software einzuordnen, aber dennoch ein Spiel, das man immer wieder gern laden wird.



Super Frogger für den TI-99/4A (Ext. Basic)

Das beliebte Computerspiel um die Rettung des kleinen Frosches, der sicher und wohlbehalten über eine befahrene Straße zu seinen Laichgründen geführt werden soll, beginnt in der vorliegenden Version mit der interessanten Status-Abfrage, ob Sie das Spiel in Stufe II (für Fortgeschrittene) spielen wollen oder sich mit der Anfängerstufe bescheiden: Der Clou ist allerdings, daß Sie Stufe II nur spielen können, wenn Stufe I zuvor gemeistert worden ist.
Der Zeitparameter, der zur Kontrolle Ihrer Leistung unerläßlich ist, wird grafisch durch einen immer kürzer werdenden Balken in Rot kenntlich gemacht.
Darüber hinaus ist die SUPER-FROGGER-Version reizvoll durch eine ordentliche Bildschirmgrafik mit vielerlei Effekten. Vier Leben besitzt Ihr Frosch - und ungeachtet der Tatsache, daß er hin und wieder auch das eine oder andere verlieren kann, ist die Sache sehr amüsant...

Zwei neue Programme vom Piman: Pi-Balled und Olympimania für den Spectrum 48K

Wenn man schon einmal einen Gag hat, dann muß man ihn auch voll ausreizen. Nach diesem Motto handelt der englische Softwarehersteller Automata und bringt ständig neue Abenteuer seines Comikhelden Piman auf den Softwaremarkt.

Pi-Balled:

Man stelle sich eine geheimnisvolle Pyramide in einer öden Wüstenlandschaft, ein kleines rosarotes Männchen mit einer großen Nase vor, und hat damit die neue Episode aus der Piman-Ära vor sich. Um den Sinn des Lebens zu erforschen, hat sich die zentrale Figur dieses Spiels, hier Burt genannt, zur Meditation in die Wüste zurückgezogen. Nach einigen Tagen, in denen er in den Sanddünen umherirrt, entdeckt er die legendäre und geheimnisvolle Pyramide von Pi. Auf dieses alte Bauwerk klettert Burt, um zu meditieren. Doch er hat die Rechnung ohne den Piman und seine Gehilfen, die Ballbrothers gemacht, die überall umher geistern. Der arme Burt muß also auf der Pyramide umherspringen, deren einzelne Segmente ständig ihre Farben verändern. Dabei wird er von den Ballbrothers, einer Schlange und verschiedenen anderen Fabelwesen verfolgt. Durch geschicktes Aufspringen auf sich drehende Scheiben kann er seinen Feinden entkommen.
Diese Geschichte ist zwar reichlich weit hergeholt,

Monster Fruit

für den TI-99/4A + Ext. Basic

Die Spielfigur, Ritter Kuno, hat die Aufgabe, Kirschen zu pflücken. Dies geschieht, indem der Hebel des Joysticks 2 nach oben gelegt wird. Jedoch wird sein eifriges Pflücken von Ungeheuern gestört. Ritter Kuno muß also über die Ungeheuer springen. Dies geschieht, indem der Fire Button des Joysticks gedrückt und gleichzeitig die Richtung, in welche Ritter Kuno springen soll, mit dem Hebel des Joysticks bestimmt wird.

SPRUNG LINKS:
Betätigen des Fire Button und gleichzeitig Hebel des Joysticks nach links.

SPRUNG RECHTS:
Betätigen des Fire Button und gleichzeitig Hebel des Joysticks nach rechts.

Hat Ritter Kuno alle Früchte gepflückt, so hat er seine Ritterrüstung anzulegen und sein Schwert zu holen, welches sich in der linken oberen Ecke befindet. Achtung! Springen Sie gegen einen spitzen Stein, so stirbt Ritter Kuno. Jetzt erscheint ein Monster, welches ihm die Kirschen wegfressen will. Er nimmt also sein Schwert und vernichtet nur die Monster, welche seine Kirschen fressen. Auf den Leitern kann er sie jedoch nicht erledigen.

Von Runde zu Runde wird es für ihn schwerer. Sie haben am Anfang 300 Zeiteinheiten. Sind alle Zeiteinheiten verbraucht ohne das Schwert zu holen, taucht ein Monster auf, das die Kirschen frißt.

Ritter Kuno ist auch verloren, wenn das Monster alle Kirschen in einer Etage aufgefressen hat.

Haben Sie die ersten Runden überstanden, erscheint eine neue Variante, das Bild verändert sich ein wenig. Nun müssen Sie mit einem Aufzug fahren, um die Kirschen zu pflücken. Dazu legen Sie einfach im Aufzug den Hebel des Joysticks nach oben. Achtung: Nur im fahrenden Aufzug ist Ritter Kuno vor den Monstern sicher.

Sind Sie unter die 10 Besten gekommen, tragen Sie sich bitte in die Bestenliste ein, dazu ist die "Alpha lock" Taste zu drücken, während beim normalen Spiel diese nicht gedrückt sein darf.

Viel Spaß!

```
100 DIM NAM$(10),PUH(10)
110 CALL CLEAR
120 DISPLAY AT(6,8):"******
***********"
130 DISPLAY AT(7,8):"*" ::
DISPLAY AT(7,24):"*"
140 DISPLAY AT(8,8):"* MONS
TER FRUIT *"
150 DISPLAY AT(9,8):"*" ::
DISPLAY AT(9,24):"*"
160 DISPLAY AT(10,8):"*  WR
ITTEN BY   *"
170 DISPLAY AT(11,8):"*" ::
 DISPLAY AT(11,24):"*"
180 DISPLAY AT(12,8):"*
 STEFAN    *"
190 DISPLAY AT(13,8):"*
WERTGEN    *"
200 DISPLAY AT(14,8):"*" ::
 DISPLAY AT(14,24):"*"
210 DISPLAY AT(15,8):"*****
***********" :: FOR I=1 TO
 500
220 NEXT I
230 CALL SCREEN(11):: CALL
CLEAR
240 DATA 2,2,4,9,13,6,12,7,
6,13,2,6
250 DATA 2,1,33,32,3,1,131,
128,7,1,44,64,9,1,131,128,9
,13,98,2,9,15,98,2,13,1,44
,64,15,1,34,32
260 DATA 16,1,33,32,17,1,13
1,128,21,1,44,64,23,1,34,32
,17,7,98,2,17,23,98,2
270 DATA 7,11,13,19,27
280 DATA 3,4,9,4,15,6
290 DATA 3,11,9,15,17,9,21,
31,9,3,5,17,9,21,17,25,31,1
7
300 DATA 3,3,17,7,9,17,13,1
5,17,19,23,17,27,31,17,3,3,
9,7,19,9,23,23,9,27,31,9
310 FOR I=1 TO 4 :: READ A,
B,C :: CALL COLOR(A,B,C)::
NEXT I :: HI=0
320 CALL CHAR(36,"FF0000000
000000000000000000000FFFF00
000000000000000000000000
FF")
330 CALL CHAR(40,"0103070F1
F3F7FFF80C0E0F0F8FCFEFF")
340 CALL CHAR(35,"0101010
1010101")
350 CALL CHAR(58,"FFFF60606
0606060FFFF060606060606")::
 CALL CHAR(128,"FFFF606060
606060FFFF060606060606")::
CALL CHAR(131,"00")
360 CALL CHAR(96,"FFFF01010
1193D03FFFF8080B09B80C0")::
 CALL CHAR(120,"03070F1F1F
0F0703C0E0F0FBF8F0E0C0")
370 CALL CHAR(98,"FFFF")::
CALL CHAR(42,"FFFF606060606
060FFFF060606060606")):: CA
LL CHAR(44,"00")
380 CALL CHAR(132,"FFFF7F7F
3F3F1F1F0F0F07070303010 1FFF
FFEFEFCFCF8F8F0F0E0E0C0C08
080")
390 CALL CHAR(34,"FFFF")::
CALL CHAR(33,"000000000000F
FFF")):: CALL CHAR(130,"202
0207020")
400 CALL CHAR(104,"00030735
37333F1F070F0F070706060E00C
0E0A0E0C0F8FCECFCFCECE0606
070")
410 CALL CHAR(100,"20232775
37333F1F070F0F070706060E00C
0E0A0E0C0F8FCECFCFCECE0606
070")
420 CALL CHAR(112,"0043261F
7F030307000000000000000008
0F9FDFEF8183800000000000
000")
430 CALL CHAR(116,"00000000
00000000043261F7F030307000
00000000000000080F9FDFEF81
833")
440 CALL CLEAR :: BILD=1 ::
 M=0 :: P=0 :: CALL MAGNIFY
(3):: R=0 :: QR=0 :: DISPL
AY AT(1,3):"PUNKTE ":P :: D
ISPLAY AT(1,18):"HISC ":HI
450 CALL DELSPRITE(ALL):: R
ESTORE 250 :: FOR I=1 TO 14
 :: READ A,B,C,D :: CALL H
CHAR(A,B,C,D):: NEXT I
460 FOR I=1 TO 5 :: READ A
:: CALL HCHAR(3,A,132):: CA
LL HCHAR(3,A+1,134):: CALL
 HCHAR(4,A,133):: CALL HCHA
R(4,A+1,135):: NEXT I
470 GOSUB 1240
480 Q=0 :: V=0 :: W=0 :: X1
=113 :: Y1=161 :: PX=15 ::
PY=21 :: L=0 :: CALL SPRIT
E(#1,104,5,Y1,X1):: IF BILD
=2 THEN 530
490 IF R=0 THEN K=13 ELSE G
OSUB 1200 :: GOSUB 1220 ::
H=INT(RND*241)+1 :: CALL S
PRITE(#6,116,2,137,H,0,11):
: CALL HCHAR(15,13,32,2)::
K=26
500 IF R>4 THEN H=INT(RND*2
41)+1 :: CALL SPRITE(#7,112
,2,105,H,0,-10):: H=INT(RN
D*241)+1 :: CALL SPRITE(#8,
116,2,25,H,0,11)
510 H=INT(RND*241)+1 :: CAL
L SPRITE(#4,112,2,57,H,0,-1
0,#2,112,2,169,241,0,-10,#
3,116,2,73,1,0,11)
520 CALL HCHAR(3,5,130):: Q
Z=5 :: W2=300 :: GOTO 630
530 RESTORE 280 :: FOR I=1
TO 3 :: READ A,B :: CALL VC
```

```
HAR(A,5,35,B):: CALL VCHAR
(A+B,5,32,2)
540 CALL VCHAR(A,25,35,D)::
 CALL VCHAR(A+B,25,32,2)::
NEXT I
550 FOR J=6 TO 26 STEP 20 ::
: CALL VCHAR(3,J,32,20):: C
ALL HCHAR(21,J-1,36):: CAL
L HCHAR(22,J-1,37,2):: CALL
 HCHAR(21,J,38):: NEXT J
560 CALL HCHAR(22,11,40)::
CALL HCHAR(22,12,41):: CALL
 HCHAR(22,17,40):: CALL HC
HAR(22,18,41)
570 CALL HCHAR(17,11,98,2)::
: CALL HCHAR(17,17,98,2)::
CALL HCHAR(9,21,98,2):: CA
LL HCHAR(3,9,130):: QZ=9
580 H=INT(RND*241)+1 :: CAL
L SPRITE(#2,112,2,105,H,0,1
0):: H=INT(RND*241)+1 :: C
ALL SPRITE(#3,112,2,57,H,0,
-10):: W2=350
590 IF R>4 THEN H=INT(RND*2
41)+1 :: CALL SPRITE(#7,116
,2,137,H,0,-11):: H=INT(RN
D*241)+1 :: CALL SPRITE(#8,
116,2,25,H,0,11)
600 IF R>4 THEN CALL HCHAR(
15,21,32,2)
610 IF R>2 THEN H=INT(RND*2
41)+1 :: CALL SPRITE(#6,116
,2,73,H,0,-11):: CALL SPRI
TE(#4,112,2,169,1,0,10):: G
OTO 630
620 CALL COINC(ALL,Z):: IF
Z=-1 THEN 1270
630 W=W+1 :: IF W>15 AND V=
1 THEN 950 ELSE IF W>W2 THE
N 930
640 CALL KEY(2,H,S):: CALL
JOYST(2,X2,Y2):: ON BILD GO
TO 650,660
650 ON PY-6 GOTO 820,670,67
0,670,670,670,780,670,670,6
70,670,670,670,670,680
660 IF PY=7 THEN 1670 ELSE
IF PY=13 THEN 1550 ELSE IF
PY=21 THEN 1510 ELSE 1610
670 Y1=Y1-2*Y2 :: PY=PY-Y2/
4 :: CALL LOCATE(#1,Y1,X1):
: CALL COINC(ALL,Z):: IF Z
=-1 THEN 1270 ELSE 630
680 IF H=18 THEN IF X2=0 TH
EN 620 ELSE CALL SPRUNG(X2,
PX,PY,X1,Y1,T):: IF T=-1 T
HEN 1270 ELSE 630
690 IF Y2=4 THEN IF (PX<=7 O
R PX=23)AND R>0 THEN 670 EL
SE 710
700 CALL KONZ(X2,PX,X1,Y1,T
):: IF T=-1 THEN 1270 ELSE
630
710 CALL SOUND(-50,210,4)::
 Y1=Y1-16 :: CALL LOCATE(#1
,Y1,X1):: CALL COINC(ALL,Z
):: IF Z=-1 THEN 1270
720 Y1=Y1-16 :: CALL LOCATE
(#1,Y1,X1):: IF V=1 THEN 10
10
730 CALL GCHAR(PY-4,PX,A)::
 IF A<>96 THEN 760
740 FOR DUR=1000 TO 800 STE
P -100 :: CALL SOUND(50,110
+DUR,3):: NEXT DUR :: CALL
 HCHAR(PY-4,PX,98,2):: CALL
 HCHAR(PY-3,PX,131,2):: P=P
+50
750 DISPLAY AT(1,9)SIZE(7):
USING "######":P :: K=K-1 :
: IF K=0 AND R<2 AND BILD=
1 THEN R=R+1 :: ON R GOSUB
1200,1220
760 CALL SOUND(-100,-4,3)::
 FOR I=1 TO 2 :: Y1=Y1+16 :
: CALL LOCATE(#1,Y1,X1)::
CALL COINC(ALL,Z):: IF Z=-1
 THEN 1270
770 NEXT I :: GOTO 630
780 IF H=18 THEN IF X2=0 TH
EN 620 ELSE CALL SPRUNG(X2,
PX,PY,X1,Y1,T):: IF T=-1 T
HEN 1270 ELSE 810
790 IF Y2=4 THEN IF PX=19 A
ND R>1 THEN 670 ELSE 710
800 IF Y2=-4 AND(P)=7 OR PX
=23)THEN 670 ELSE CALL KONZ
(X2,PX,X1,Y1,T):: IF T=-1
THEN 1270
810 IF X1>88 AND X1<112 AND
 R>3 THEN 1270 ELSE 630
820 IF H=18 THEN IF X2=0 TH
EN 620 ELSE CALL SPRUNG(X2,
PX,PY,X1,Y1,T):: IF T=-1 T
HEN 1270 ELSE 630
830 IF Y2=-4 AND PX=19 THEN
 670
840 IF Y2<>4 THEN CALL KONZ
(X2,PX,X1,Y1,T):: IF T=-1 T
HEN 1270 ELSE 630
850 CALL SOUND(-50,220,3)::
 Y1=Y1-16 :: CALL LOCATE(#1
,Y1,X1):: CALL COINC(ALL,Z
):: IF Z=-1 THEN 1270
860 Y1=Y1-16 :: CALL LOCATE
(#1,Y1,X1):: CALL GCHAR(3,P
X,A):: CALL GCHAR(3,PX+1,B
):: IF (A=131 AND B=131)OR(
B=130 AND BILD=1)THEN 760
870 IF A<>130 THEN 1270
880 CALL COLOR(#1,2):: CALL
 PATTERN(#1,108):: CALL HCH
AR(3,QZ,131):: Q=1
890 IF M>0 OR K>0 THEN DISP
LAY AT(24,1):"  SORRY NO BO
NUS" :: GOTO 910
900 PU=(W2-W)*(R+3):: DISPL
AY AT(24,1):"TIME ":W :: DI
SPLAY AT(24,11):"BONUS ":P
U :: P=P+PU :: DISPLAY AT(1
,9)SIZE(7):USING "######":P
910 IF K=0 THEN GOSUB 1240
920 IF M=0 THEN 930 ELSE W=
0 :: GOTO 760
930 PTX=-1 :: W=0 :: X5=1 :
: V=1 :: M=R+1 :: PTT=16 ::
 H=INT(RND*2):: IF H=0 THE
N PTT=-16 :: X5=257 :: PTX=
35
940 IF Q=1 THEN 760 ELSE 64
0
950 W=0 :: X5=X5+PTT :: PTX
=PTX+PTT/8 :: IF L=1 THEN 9
80
960 CALL SOUND(100,110,0)::
 L=1 :: H=INT(RND*2)+1 :: P
TY=17 :: IF H=1 THEN Y5=12
9 ELSE Y5=65 :: PTY=9
970 CALL SPRITE(#5,112,2,Y5
,X5):: GOTO 640
980 CALL SOUND(100,110,2)::
 CALL GCHAR(PTY,PTX,A):: IF
 A<>96 THEN 1000
990 CALL HCHAR(PTY,PTX,98,2
):: CALL HCHAR(PTY+1,PTX,13
1,2):: K=K-1 :: IF K=0 THE
N 1270
1000 CALL SOUND(100,110,0):
: CALL LOCATE(#5,Y5,X5):: G
OTO 640
1010 CALL COINC(#1,#5,9,Z):
: IF Z<>-1 THEN 760 ELSE IF
 Q=0 THEN 1270
1020 CALL SOUND(100,220,0):
: CALL DELSPRITE(#5):: P=P+
INT(100/M)*(R+1):: DISPLAY
 AT(1,9)SIZE(7):USING "####
##":P :: M=M-1 :: W=0 :: L=
0 :: IF M<>0 THEN 760
1030 IF PTT=16 THEN PU=(257
-X5)*(R+1)ELSE PU=X5*(R+1)
1040 DISPLAY AT(24,1):"  BO
NUS ":PU :: P=P+PU :: DISPL
AY AT(1,9)SIZE(7):USING "#
######":P
1050 Q=0 :: V=0 :: K=K+1 ::
 CALL PATTERN(#1,104):: CAL
L HCHAR(3,QZ,130):: CALL C
OLOR(#1,5):: IF BILD=2 THEN
 1130
1060 IF R=3 THEN CALL SPRIT
E(#6,112,2,145,1,0,10):: GO
TO 760
1070 IF R=4 THEN CALL HCHAR
(15,13,32,2):: GOTO 760
1080 IF R=5 AND QR=0 THEN Q
R=1 :: BILD=2 :: R=2 :: GOT
O 450
1090 IF R=5 THEN CALL SPRIT
E(#7,112,2,105,241,0,-10)::
 GOTO 760
1100 IF R=6 THEN CALL SPRIT
E(#8,116,2,25,1,0,11):: GOT
O 760
1110 IF R=8 AND QR=1 THEN B
ILD=2 :: QR=0 :: R=4 :: GOT
O 450
1120 IF R=8 THEN BILD=2 ::
R=7 :: GOTO 450
1130 IF R=3 THEN CALL SPRIT
E(#4,112,2,169,1,0,10):: GO
TO 760
1140 IF R=4 THEN CALL SPRIT
E(#6,116,2,73,241,0,-11)::
GOTO 760
1150 IF R=5 AND QR=1 THEN B
ILD=1 :: R=4 :: GOTO 450
1160 IF R=5 THEN CALL SPRIT
E(#7,116,2,137,241,0,-11)::
 GOTO 760
1170 IF R=6 THEN CALL HCHAR
(15,21,32,2):: GOTO 760
1180 IF R=7 THEN CALL SPRIT
E(#8,116,2,25,1,0,11):: GOT
O 760
1190 IF R=8 THEN BILD=1 ::
R=7 :: GOTO 450
1200 K=13 :: G=7 :: FOR I=1
 TO 2 :: CALL VCHAR(14,G,42
):: CALL VCHAR(14,G+1,43):
: CALL VCHAR(15,G,58,2):: C
ALL VCHAR(15,G+1,59,2)
1210 CALL VCHAR(17,G,128,4)
:: CALL VCHAR(17,G+1,129,4)
:: CALL VCHAR(21,G,42,2)::
 CALL VCHAR(21,G+1,43,2)::
G=23 :: NEXT I :: RETURN
1220 CALL VCHAR(9,19,128,4)
:: CALL VCHAR(9,20,129,4)::
 CALL VCHAR(8,19,42):: CAL
L VCHAR(8,20,43)
1230 CALL VCHAR(13,19,42,2)
:: CALL VCHAR(13,20,43,2)::
 RETURN
1240 RESTORE 290 :: TT=6 ::
 K=26 :: IF BILD=2 THEN RES
TORE 300 :: TT=9 :: K=23
1250 FOR I=1 TO TT :: READ
A,B,C :: FOR J=A TO B STEP
2 :: CALL HCHAR(C,J,96)::
CALL HCHAR(C,J+1,97):: CALL
 HCHAR(C+1,J,120):: CALL HC
HAR(C+1,J+1,121)
```

```
1260 NEXT J :: NEXT I :: RE
TURN
1270 CALL DELSPRITE(ALL)
1280 FOR T=15 TO 0 STEP -1
:: CALL SOUND(-50,(T*85)+1:
0,2*T):: NEXT T
1290 CALL CLEAR :: CALL DEL
SPRITE(ALL):: IF P>HI THEN
HI=P :: CALL HIGH
1300 CALL DELSPRITE(ALL)
1310 DISPLAY AT(5,3):"SIE H
ABEN VERLOREN"
1320 DISPLAY AT(10,3):"SIE
HABEN ":P :: DISPLAY AT(11,
3):"PUNKTE ERREICHT"
1330 IF P>15000 THEN DISPLA
Y AT(13,3):"DU BIST DER CHA
MPION" :: GOTO 1390
1340 IF P>10000 THEN DISPLA
Y AT(13,3):"EIN GROSSARTIGF
S ERGEBNIS" :: GOTO 1390
1350 IF P>5000 THEN DISPLAY
 AT(13,3):"SCHON GANZ GUT"
:: GOTO 1390
1360 IF P>2000 THEN DISPLAY
 AT(13,3):"SIE MUESSEN NOCH
" :: DISPLAY AT(15,3):"ETW
AS UEBEN" :: GOTO 1390
1370 IF P>1000 THEN DISPLAY
 AT(13,3):"NOCH SEHR ERBAER
MLICH" :: GOTO 1390
1380 DISPLAY AT(13,3):"SIE
SIND EIN" :: DISPLAY AT(15,
3):"BLUTIGER ANFAENGER"
1390 IF PUH(10)>P THEN MAN=
0 :: GOTO 1430
1400 PUH(10)=P :: FOR I=9 T
O 1 STEP -1 :: IF PUH(I)>PU
H(I+1) THEN MAN=I+1 :: GOTO
 1420
1410 MAN=PUH(I):: NAM$(I+1)
=NAM$(I):: PUH(I)=PUH(I+1):
: PUH(I+1)=MAN :: NEXT I :
: MAN=1
1420 NAM$(MAN)=""
1430 DISPLAY AT(17,3):"BITT
E TASTE DRUECKEN " :: CALL
KEY(0,K,S):: IF S=0 THEN 1
430
1440 CALL CLEAR
1450 FOR I=1 TO 10 :: DISPL
AY AT(I+2,3):USING "##":I :
: DISPLAY AT(I+2,6):NAM$(I
):: DISPLAY AT(I+2,12)SIZE(
7):USING "######":PUH(I)::
DISPLAY AT(I+2,20):"PKT" :
: NEXT I
1460 DISPLAY AT(15,3):"IHR
ERGEBNIS IST ":P :: IF MAN=
0 THEN 1500
1470 DISPLAY AT(17,3):"SIE
SIND AUF PLATZ ":MAN
1480 DISPLAY AT(19,3):"GERF
N SIE IHREN " :: DISPLAY AT
(21,3):"NAMEN EIN"
1490 ACCEPT AT(MAN+2,6)SIZE
(6):NAM$(MAN):: MAN=0
1500 DISPLAY AT(23,3):"BITT
E TASTE DRUECKEN" :: CALL K
EY(0,K,S):: IF S=0 THEN 15
00 ELSE GOTO 440
1510 IF H=18 THEN IF X2=0 T
HEN 620 ELSE CALL SPRUNG(X2
,PX,PY,X1,Y1,T):: IF T=-1
THEN 1270 ELSE 1540
1520 IF Y2=4 THEN CALL GCHA
R(PY,PX,A):: IF A=36 THEN 1
600 ELSE 710
1530 CALL KONZ(X2,PX,X1,Y1,
T):: IF T=-1 THEN 1270
1540 IF (PX>9 AND PX<13)OR(
PX>15 AND PX<19)THEN 1270 E
LSE 630
1550 IF H=18 THEN IF X2=0 T
HEN 620 ELSE CALL SPRUNG(X2
,PX,PY,X1,Y1,T):: IF T=-1
THEN 1270 ELSE 1580
1560 CALL GCHAR(PY,PX,A)::
IF Y2=4 THEN IF A=36 THEN 1
600 ELSE 710
1570 IF Y2=-4 AND A=36 THEN
 1600 ELSE CALL KONZ(X2,PX,
X1,Y1,T):: IF T=-1 THEN 12
70
1580 IF R>5 AND PX>19 AND P
X<23 THEN 1270
1590 CALL GCHAR(PY,PX,A)::
CALL GCHAR(PY,PX,B):: IF A=
32 OR B=32 THEN 1270 ELSE
630
1600 CALL SPRITE(#10,36,2,Y
1,X1):: CALL HCHAR(PY,PX,32
,2):: CALL HCHAR(PY+1,PX,3
2,2):: PY=0
1610 IF Y2=0 THEN 1630 :: C
ALL MOTION(#10,-Y2*2,0,#1,-
Y2*2,0)
1620 CALL POSITION(#1,Y1,A)
:: IF Y1>169 OR Y1<8 THEN 1
270 ELSE 630
1630 CALL MOTION(#10,0,0,#1
,0,0):: CALL POSITION(#1,Y1
,A)
1640 IF (Y1>136 AND Y1<165)
OR(Y1>72 AND Y1<101)OR(Y1>4
4 AND Y1<52)THEN 1650 ELSE
 630
1650 PY=INT((Y1+3)/8+1):: Y
1=(PY-1)*8+1 :: CALL LOCATE
(#1,Y1,X1)
1660 CALL DELSPRITE(#10)::
CALL HCHAR(PY,PX,36):: CALL
 HCHAR(PY+1,PX,37,2):: CAL
L HCHAR(PY,PX+1,38):: GOTO
630
1670 IF H=18 THEN IF X2=0 T
HEN 620 ELSE CALL SPRUNG(X2
,PX,PY,X1,Y1,T):: IF T=-1
THEN 1270 ELSE 1590
1680 IF Y2=-4 THEN CALL GCH
AR(PY,PX,A):: IF A=36 THEN
1600
1690 IF Y2=4 THEN 050 ELSE
CALL KONZ(X2,PX,X1,Y1,T)::
IF T=-1 THEN 1270 ELSE 159
0
1700 SUB SPRUNG(X2,PX,PY,X1
,Y1,T)
1710 PX=PX+X2 :: IF PX<1 OR
 PX>32 THEN T=-1 :: IF T=-1
 THEN SUBEXIT
1720 CALL SOUND(-50,-2,4)::
 FOR J=-4 TO 4 STEP 8 :: FO
R I=1 TO 4 :: X1=X1+X2 ::
Y1=Y1+J :: CALL LOCATE(#1,Y
1,X1)
1730 CALL COINC(ALL,T):: IF
 T=-1 THEN SUBEXIT
1740 NEXT I :: NEXT J :: SU
BEND
1750 SUB KONZ(X2,PX,X1,Y1,T
)
1760 PX=PX+X2/4 :: IF PX<1
OR PX>32 THEN T=-1 :: SUBEX
IT
1770 X1=X1+2*X2 :: CALL LOC
ATE(#1,Y1,X1):: CALL COINC(
ALL,T):: SUBEXIT
1780 SUBEND
1790 SUB HIGH
1800 CALL SCREEN(2):: CALL
MAGNIFY(2)
1810 CALL SPRITE(#1,72,16,6
5,84,#2,73,16,49,108,#3,71,
16,49,132,#4,72,16,65,156)
1820 CALL SPRITE(#5,83,16,1
05,72,#6,67,16,121,96,#7,79
,16,137,120,#8,82,16,121,1
44,#9,69,16,105,168)
1830 FOR I=1 TO 10 :: CALL
SCREEN(2):: CALL SOUND(100,
440,0,523,0):: FOR J=1 TO
100 :: NEXT J :: CALL SCREE
N(16):: FOR J=1 TO 10 :: NE
XT J :: NEXT J
1840 CALL SCREEN(11):: CALL
 CLEAR :: SUBEND
```

SKI

für den TI-99/4A

"Ski" ist ein Spiel für den TI-99/4A, in dem es darum geht, einen Skiläufer verschiedene Pisten hinunterfahren zu lassen. Dabei macht es am meisten Spaß, wenn man das Spiel mit mehreren Personen spielt: bis zu 8 Spieler können dran teilnehmen. So ist es z.B. möglich, innerhalb der Familie oder des Bekanntenkreises ein kleines Meisterschaftsrennen durchzuführen.
Um dieses Spiel zu spielen, ist nur die Grundversion erforderlich und eventuell Joysticks. Läßt sich aber auch mit der Tastatur meistern.
Zu Beginn des Spieles fragt der Computer, wieviele Spieler teilnehmen wollen. Danach müssen die Namen der Mitspieler eingegeben werden.
Nach der Ankündigung des ersten Läufers erscheint das Startbild. Der Skiläufer läßt sich nun durch Drücken der Feuertaste oder einer beliebigen anderen Taste starten. In der rechten oberen Ecke des Bildes wird die Startfrist angezeigt.
Läuft die Zeit ab, ohne daß der Läufer gestartet wurde, wird er disqualifiziert.
Nach dem Start läßt sich der Skiläufer mit Hilfe der Fernbedienung (bzw. der Tasten 'S' und 'D') nach links und rechts bewegen. Die Feuertaste (bzw. die Taste 'Y') läßt den Skiläufer schneller fahren.
Hierbei ist jedoch zu beachten, daß er nicht nach links oder rechts bewegt wird, da er sonst stürzen würde.
Auf der Piste befinden sich als Hindernisse Tannen und Felsbrocken, die umfahren werden müssen. Fährt der Läufer auf ein solches Hindernis auf, stürzt er.
Beim Überfahren der Stangen passiert weiter nichts, jedoch zählt am Schluß jedes nicht durchfahrene Tor fünf Strafsekunden.
Erreicht der Skiläufer das Ziel, erfolgt die Angabe der benötigten Zeit, die Anzahl der nicht durchfahrenen Tore sowie die daraus resultierende Gesamtzeit.
Zum Schluß erfolgt die Auswertung, in der die Ergebnisse aller Spieler in der Reihenfolge ihrer Plätze angezeigt werden.
Insgesamt müssen drei verschiedene Pisten durchfahren werden. Jeder Läufer hat dazu eine eigene Skianzugfarbe. Derjenige, der die geringste Zeit benötigt, ist "Familienmeister" (o.ä.)?

Spielaufbau:
100 - 610 Zeichendefinitionen
620 - 820 Titel
830 - 1000 1. Spielfeld einlesen und Eingabe der Spieler
1040 - 1070 Abfrage, ob Spieler disqualifiziert

```
100 REM  SKIFAHRER
110 REM  14 KBYTE
120 REM    BY
 .         EDGAR KLOEPFEL
           QUITTFRBLICK
3               3501 GUXHAGEN
-                       BUECHENW
FRRA
130 CALL CLEAR
140 CALL SCREEN(16)
150 FOR I=5 TO 8
160 CALL COLOR(I,1,1)
170 NEXT I
180 CALL COLOR(10,1,1)
190 DIM P(3,8)
200 DIM S$(120)
210 DIM SM(120)
220 CALL CHAR(101,"4020508B0
48040FC")
230 CALL CHAR(102,"00E0105F7
F6F18E1")
240 CALL CHAR(103,"02040A112
001023F")
250 CALL CHAR(96,"000708F0FE
7608C7")
260 CALL CHAR(100,"001C1C0B3
E5D5D1C")
270 CALL CHAR(98,"5563414141
4141")
280 CALL CHAR(99,"1C2C342546
B40810")
290 CALL CHAR(97,"3C342CA462
211008")
300 CALL CHAR(128,"000103070
D030700")
310 CALL CHAR(129,"008030E0B
0D0E0B0")
320 CALL CHAR(130,"13070D1B2
70D193.")
330 CALL CHAR(131,"C8E0D0D8E
4B098BC")
340 CALL CHAR(104,"410101030
70D1900")
350 CALL CHAR(105,"B2B080C0E
0B09B00")
360 CALL CHAR(136,"000000000
1070F0F")
370 CALL CHAR(137,"00000000E
0F0FCFC")
380 CALL CHAR(138,"1F3F3F7F7
F7F1C00")
390 CALL CHAR(139,"FEFCFEFEF
EFE3C00")
400 CALL CHAR(144,"040C13040
4040404")
410 CALL CHAR(145,"10:81C101
0101010")
420 CALL CHAR(146,"040404040
40E")
430 CALL CHAR(147,"101010101
038")
440 CALL CHAR(132,"00102C5E7
65A343C")
450 CALL CHAR(133,"00102C5E7
65A343C")
460 CALL CHAR(42,"")
470 CALL CHAR(120,"FFFFFFFFF
FFFFFFF")
480 CALL CHAR(36,"030303030303
030303")
490 CALL CHAR(37,"C0C0C0C0C0
C0C0C0")
500 CALL CHAR(38,"0000000000
0000FF")
510 CALL CHAR(35,"0101010101
010101")
520 CALL CHAR(34,"8080808080
808080")
530 CALL CHAR(39,"FF")
540 CALL CHAR(43,"4400444444
444438")
550 CALL CHAR(64,"304492A2A2
924438")
560 CALL CHAR(62,"0082384444
7C4444")
570 CALL CHAR(111,"44007C444
444447C")
580 CALL CHAR(92,"")
590 CALL CHAR(93,"")
600 CALL COLOR(12,1,1)
610 CALL VCHAR(1,3,120,24*28
)
620 PRINT "XXXXXXXX   X XXX
X XXXXXXXX"
630 PRINT "XXXXXXX XXXX XXX
X XXXXXXX"
640 PRINT "XXXXXXX XXXX XX
XX XXXXXXX"
650 PRINT "XXXXXXX XXXX X X
XX XXXXXXX"
660 PRINT "XXXXXXXX  XX  XX
XX XXXXXXX"
670 PRINT "XXXXXXXXXXX X X X
XX XXXXXXX"
680 PRINT "XXXXXXXXXXX X XX
XX XXXXXXX"
690 PRINT "XXXXXXXXXXX X XXX
X XXXXXXX"
700 PRINT "XXXXXXX   XX XXX
X XXXXXXX":::::::
710 FOR I=1 TO 6
720 CALL HCHAR(17+I,3,120,28
)
730 NEXT I
740 PRINT ":::@xDVxEDGARxKLoP
FEL(x2/B4xx"
750 CALL HCHAR(24,3,120,28)
760 CALL COLOR(12,7,1)
770 FOR I=2 TO 8
780 CALL COLOR(I,16,7)
790 NEXT I
800 CALL COLOR(10,16,7)
810 CALL COLOR(11,15,1)
820 CALL COLOR(13,3,1)
830 RESTORE 2550
840 GOSUB 4140
850 CALL CLEAR
860 FOR I=2 TO 8
870 CALL COLOR(I,2,1)
880 NEXT I
890 CALL COLOR(10,9,1)
900 PRINT "ANZAHL DER TEILNE
HMER?":'        (MAX. 8)"::::
910 INPUT ANZ
920 IF ANZ<1 THEN 910
```

```
930 IF ANZ>8 THEN 910
940 CALL CLEAR
950 PRINT "BITTE GEBEN SIE D
IE NAMEN   ALLER L>UFER EIN!
"::::
960 FOR SP=1 TO ANZ
970 ER(SP)=ER(SP)+6000+SP*.0
01
980 PRINT "L>UFER";SP
990 INPUT L$(SP)
1000 NEXT SP
1010 PARC=1
1020 CALL CLEAR
1030 SP=1
1040 ER(SP)=ER(SP)-600
1050 IF ER(SP)<11000 THEN 10
80
1060 SP=SP+1
1070 IF SP>ANZ THEN 3580 ELS
E 1050
1080 ON SP GOSUB 1110,1130,1
150,1170,1190,1210,1230,1250
1090 CALL COLOR(9,COL,1)
1100 GOTO 1270
1110 COL=5
1120 RETURN
1130 COL=7
1140 RETURN
1150 COL=13
1160 RETURN
1170 COL=2
1180 RETURN
1190 COL=11
1200 RETURN
1210 COL=14
1220 RETURN
1230 COL=8
1240 RETURN
1250 COL=4
1260 RETURN
1270 Z=0
1280 PRINT TAB(6);"AM START
L>UFER";SP;:TAB(8);L$(SP);::
:::::::
1290 FOR M=1 TO 1000
1300 NEXT M
1310 CALL CLEAR
1320 T=0
1330 TOR=10
1340 F=98
1350 A=17
1360 A$="LAUF "&STR$(PARC)
1370 VER=1
1380 HOR=2
1390 GOSUB 4210
1400 A$="L>UFER '&STR$(SP)
1410 VER=3
1420 GOSUB 4210
1430 CALL VCHAR(2,A-3,35,5)
1440 CALL VCHAR(2,A+3,34,5)
1450 CALL HCHAR(3,A-2,39,5)
1460 CALL HCHAR(1,A-2,38,5)
1470 CALL HCHAR(2,A-2,83)
1480 CALL HCHAR(2,A-1,84)
1490 CALL HCHAR(2,A,65)
1500 CALL HCHAR(2,A+1,82)
1510 CALL HCHAR(2,A+2,84)
1520 FOR I=1 TO 11
1530 CALL HCHAR(6+I,A-2-I,13
2)
1540 CALL HCHAR(6+I,A+2+I,13
3)
1550 NEXT I
1560 CALL VCHAR(18,4,133,7)
1570 CALL VCHAR(18,30,132,7)
1580 CALL HCHAR(5,A,100)
1590 CALL HCHAR(6,A,98)
1600 A$="ZUM STARTEN TASTE D
R+CKEN"
1610 VER=23
1620 HOR=4
1630 GOSUB 4210
1640 FOR TS=20 TO 0 STEP -1
1650 FOR I=1 TO 50
1660 NEXT I
1670 IF TS>9 THEN 1700
1680 CALL HCHAR(1,28,32)
1690 GOTO 1710
1700 CALL HCHAR(1,28,ASC(SEG
$(STR$(TS),1,1)))
1710 CALL HCHAR(1,29,ASC(SEG
$(STR$(TS),LEN(STR$(TS)),1))
)
1720 CALL KEY(1,K,S)
1730 IF S<>0 THEN 1800
1740 CALL KEY(2,K,S)
1750 IF S<>0 THEN 1800
1760 CALL KEY(0,K,S)
1770 IF S<>0 THEN 1800
1780 NEXT TS
1790 GOTO 3790
1800 CALL HCHAR(23,5,32,25)
1810 CALL VCHAR(4,A-98+F,32,
2)
1820 CALL VCHAR(5,A,100)
1830 CALL VCHAR(6,A,F)
1840 IF S<>1 THEN 1860
1850 CALL SOUND(-250,-5,25)
1860 CALL KEY(1,K,S)
1870 CALL KEY(2,E,0)
1880 IF S=-1 THEN 2030
1890 IF S=0 THEN 2010
1900 T=T+.13
1910 FOR M=1 TO 30
1920 NEXT M
1930 IF K=2 THEN 1950
1940 IF K=3 THEN 1980
1950 A=A-1
1960 F=99
1970 GOTO 2040
1980 A=A+1
1990 F=97
2000 GOTO 2040
2010 F=98
2020 GOTO 2040
2030 A=A+93-F
2040 T=T+.67
2050 IF E<>18 THEN 2080
2060 IF F=98 THEN 2080
2070 CALL HCHAR(7,A,93+(F=99
))
2080 GOSUB 3850
2090 IF E<>18 THEN 1870
2100 T=T+1.47
2110 GOSUB 3850
2120 GOSUB 3850
2130 GOSUB 3850
2140 GOTO 1840
2150 S$(I)=CHR$(144)&CHR$(32
)&CHR$(145)
2160 S$(I+1)=CHR$(146)&CHR$(
42)&CHR$(147)
2170 S$(I+2)="
2180 RETURN
2190 S$(I)=CHR$(136)&CHR$(13
7)
2200 S$(I+1)=CHR$(138)&CHR$(
139)
2210 S$(I+2)="
2220 RETURN
2230 S$(I)=CHR$(128)&CHR$(12
9)
2240 S$(I+1)=CHR$(130)&CHR$(
131)
2250 S$(I+2)=CHR$(104)&CHR$(
105)
2260 RETURN
2270 S$(I)=CHR$(136)&CHR$(13
7)&" "&CHR$(136)&CHR$(137)
2280 S$(I+1)=CHR$(138)&CHR$(
139)&" "&CHR$(138)&CHR$(139)
2290 S$(I+2)=""
2300 RETURN
2310 S$(I)=CHR$(128)&CHR$(12
9)&"      "&CHR$(128)&CHR$(12
9)
2320 S$(I+1)=CHR$(130)&CHR$(
131)&"      "&CHR$(130)&CHR$(
131)
2330 S$(I+2)=CHR$(104)&CHR$(
105)&"      "&CHR$(104)&CHR$(
105)
2340 RETURN
2350 S$(I)="$XXXXXXXXXXXXXX
%"
2360 S$(I+1)="$X   X X   X X
XX%"
2370 S$(I+2)="$XXX X X XXX X
XX%"
2380 RETURN
2390 S$(I)="$XX XX X  XX XXX
%"
2400 S$(I+1)="$X XXX X XXX X
XX%"
2410 S$(I+2)="$X   X X   X
X%"
2420 RETURN
2430 S$(I)="$XXXXXXXXXXXXXX
%"
2440 S$(I+1)="$
  %"
2450 S$(I+2)="$
  %"
2460 RETURN
2470 S$(I)=""
2480 S$(I+1)=""
2490 S$(I+2)=""
2500 RETURN
2510 S$(I)="$
%"
2520 S$(I+1)="$
  %"
2530 S$(I+2)="$
  %"
2540 RETURN
2550 DATA 1,15,2,20,3,10,1,1
8,2,12,3,10,1,11,5,17,4,06,1
,07,3,23,4,16,,,08,5,10,
2,21,2,20,5,09
2560 DATA 1,12,3,22,4,12,1,1
4,2,05,5,17,1,15,3,23,4,11,5
,15,1,13,4,08,5,09,4,14,
5,06,1,08,9,09
2570 DATA 6,07,7,07,8,07,10,
07,10,07,9,09,9,09
2580 DATA 2,08,1,12,4,21,5,0
8,5,05,1,17,2,15,3,23,1,11,5
,11
2590 DATA 1,15,2,23,3,14,1,1
8,5,12,4,06,1,11,3,15,2,13,1
,13,4,09,5,15,1,19,4,11,
3,07,5,14,1,11
2600 DATA 2,12,2,10,3,18,1,0
9,2,08,5,06,9,09,6,07,7,07,8
,07,10,07,10,07,9,09,9,0
9
2610 DATA 5,15,1,12,4,20,3,1
1,1,05,5,12,5,04,1,09,2,18,5
,12,2,14,4,09,1,17,5,05,
5,08,1,10,4,17
2620 DATA 5,06,1,07,2,05,5,1
0,1,12,4,18,4,12,1,16,5,11,1
,12,2,06,3,10,1,00,5,08,
5,12,4,05,9,09
2630 DATA 6,07,7,07,8,07,10,
07,10,07,9,09,9,09
2640 Z=Z+1
2650 CALL VCHAR(4,A+F-98,32,
2)
2660 IF PIC/2<>INT(PIC/2)THE
N 2730
2670 PRINT TAB(Z);CHR$(133);
TAB(SM(Z));S$(Z);TAB(28);CHR
$(132)
2680 CALL HCHAR(6,A-1,101)
2690 CALL HCHAR(6,A-2,102)
2700 A=A-1
2710 O=A-1
2720 GOTO 2780
2730 PRINT TAB(Z);CHR$(133);
TAB(SM(Z));S$(Z);TAB(28);CHR
$(132)
```

1080 - 1260 Läuferfarbe zuordnen
1270 - 1300 Ankündigung des Spielers am Start
1310 - 1620 Startbild
1640 - 1790 Startzeiteinblendung
1800 - 2140 eigentliches Spiel (Abfrage, Bewegen des Läufers)
2150 - 2540 Zeilenaufbau (Tore, Felsen, Tannen, Ziel usw.)
2550 - 2570 Datazeilen Spielfeld 1
2580 - 2600 Datazeilen Spielfeld 2
2610 - 2630 Datazeilen Spielfeld 3
2640 - 2870 Auswertung Zusammenstoß Läufer mit Hindernis
2880 - 3110 Ermittlung Ergebnis des Spielers und Anzeige
3120 - 3410 Auswertung mit Platzanweisung und Anzeige
3420 - 3780 Nächster Spieler, nächstes Spielfeld, Abschlußroutine
3790 - 3840 Disqualifikation eines Spielers
3850 - 3910 Abfrage, ob Tor durchfahren wurde
3920 - 3970 Abfrage, ob Läufer auf Hindernis gefahren ist
3980 - 4130 Ausgaberoutine für Zeiten (0:00.00)
4140 - 4200 Einlesen der Spielfelder
4210 - 4240 Printroutine

Zeilenänderungen in der Joystickversion

Das Spiel wurde ursprünglich als Tastaturversion geschrieben, daher sind in der Joystickversion folgende Zeilen zu ändern bzw. neu einzugeben:

```
1839 X2=
1840 IF X1=X2 THEN 1860
1859 X2=X1
1860 CALL JOYST(2,X1,Y1)
1880 IF X1=0 THEN 2010
1890 IF X1=X2 THEN 2030
1930 IF X1=-4 THEN 1950
1940 IF X1=4 THEN 1980
```

Nach dieser Eingabe kann der Skiläufer mit Joystick 2 gesteuert werden. Der Feuerknopf dient zum Starten des Läufers und um die Geschwindigkeit der Abfahrt zu erhöhen.

```
2740 CALL HCHAR(6,A+1,103)
2750 CALL HCHAR(6,A+2,96)
2760 A=A+1
2770 O=A
2780 FOR P=1 TO 1000
2790 NEXT P
2800 T=T+5.0
2810 CALL HCHAR(6,O,32,2)
2820 CALL HCHAR(4,A,100)
2830 CALL HCHAR(5,A,98)
2840 FOR P=1 TO 500
2850 NEXT P
2860 F=93
2870 GOTO 1810
2880 FOR M=1 TO 100
2890 NEXT M
2900 CALL CLEAR
2910 QM$=STR$(INT(T/60))
2920 QS$=STR$((T-INT(T/60)*6
0)+100.001)
2930 PRINT "ZEIT:"
2940 GOSUB 3980
2950 PRINT :::
2960 IF TOR<>0 THEN 2990
2970 PRINT "ALLE TORE DURCHF
AHREN.        KEINE STRAFZEIT
":::
2980 GOTO 3030
2990 IF TOR<>1 THEN 3020
3000 PRINT "STRAFZEIT F+R 1
NICHT        DURCHFAHRENES TO
R: 5 SEK.":::
3010 GOTO 3030
3020 PRINT "STRAFZEIT F+R";T
OR;"NICHT";"DURCHFAHRENE TOR
E:";5*TOR;"SEKS":::
3030 B(PARC,SP)=T+5*TOR
3040 ER(SP)=ER(SP)+B(1,SP)+B
(2,SP)+B(3,SP)
3050 QM$=STR$(INT(B(PARC,SP)
/60))
3060 QS$=STR$(B(PARC,SP)-INT
(B(PARC,SP)/60)*60+100.001)
3070 PRINT "GESAMTZEIT:"
3080 GOSUB 3980
3090 PRINT :::
3100 FOR M=1 TO 1500
3110 NEXT M
3120 CALL CLEAR
3130 A$="AUSWERTUNG"
3140 VER=1
3150 HOR=11
3160 GOSUB 4210
3170 CALL HCHAR(2,12,61,10)
3180 VER=2
3190 HOR=2
3200 FOR I=1 TO ANZ
3210 FOR J=1 TO ANZ
3220 IF ER(I)<ER(J)THEN 3240
3230 PLATZ=PLATZ+1
3240 NEXT J
3250 RANG(PLATZ)=I
3260 PLATZ=0
3270 NEXT I
3280 I=1
3290 QM$=STR$(INT(ER(RANG(I)
)/60))
3300 QS$=STR$(ER(RANG(I))-IN
T(ER(RANG(I))/60)*60+100.001
)
3310 IF ER(RANG(I))-(RANG(I)
)*.001<>ER(RANG(I-1))-(RANG(
I-1))*.001 THEN 3340
3320 A$="    "&L$(RANG(I))
3330 GOTO 3350
3340 A$=STR$(I)&". "&L$(RANG
(I))
3350 VER=VER+2
3360 GOSUB 4210
3370 IF ER(RANG(I))>=12000 T
HEN 4060
3380 GOSUB 3980
3390 I=I+1
3400 IF I>ANZ THEN 3420
3410 GOTO 3290
3420 SP=SP+1
3430 IF SP>ANZ THEN 3480
3440 FOR I=1 TO 1500
3450 NEXT I
3460 CALL CLEAR
3470 GOTO 1040
3480 PARC=PARC+1
3490 IF PARC>3 THEN 3560
3500 IF PARC=2 THEN 3530
3510 RESTORE 2610
3520 GOTO 3540
3530 RESTORE 2580
3540 GOSUB 4140
3550 GOTO 1020
3560 FOR I=1 TO 1500
3570 NEXT I
3580 FOR I=1 TO 3
3590 FOR J=1 TO 8
3600 B(I,J)=0
3610 ER(J)=0
3620 NEXT J
3630 NEXT I
3640 A$="NOCH EIN SPIEL? (J/
N)"
3650 VER=23
3660 HOR=2
3670 GOSUB 4210
3680 CALL KEY(2,K,S)
3690 IF S=0 THEN 3680
3700 IF K<>32 THEN 3770
3710 CALL CLEAR
3720 FOR I=3 TO 8
3730 CALL COLOR(I,1,1)
3740 NEXT I
3750 CALL COLOR(10,1,1)
3760 GOTO 600
3770 CALL CLEAR
3780 END
3790 CALL CLEAR
3800 PRINT "IHRE STARTZEIT I
ST LEIDER   ABGELAUFEN.":"S
IE WERDEN DISQUALIFIZIER
T!":::::::::
3810 FOR I=1 TO 1000
3820 NEXT I
3830 ER(SP)=12000+SP*.001
3840 GOTO 3120
3850 Z=Z+1
3860 PRINT TAB(2);CHR$(133);
TAB(SM(Z));S$(Z);TAB(28);CHR
$(132)
3870 CALL GCHAR(6,A,PIC)
3880 IF PIC=32 THEN 3930
3890 IF PIC<>42 THEN 3920
3900 TOR=TOR-1
3910 GOTO 3930
3920 IF PIC<144 THEN 2640
3930 IF Z>118 THEN 2880
3940 CALL VCHAR(4,A-98+F,32,
2)
3950 CALL VCHAR(5,A,100)
3960 CALL VCHAR(6,A,F)
3970 RETURN
3980 CALL HCHAR(VER,24,ASC(S
EG$(QM$,LEN(QM$),1)))
3990 CALL HCHAR(VER,25,58)
4000 CALL HCHAR(VER,26,ASC(S
EG$(QS$,2,1)))
4010 CALL HCHAR(VER,27,ASC(S
EG$(QS$,3,1)))
4020 CALL HCHAR(VER,28,46)
4030 CALL HCHAR(VER,29,ASC(S
EG$(QS$,5,1)))
4040 CALL HCHAR(VER,30,ASC(S
EG$(QS$,6,1)))
4050 RETURN
4060 CALL HCHAR(VER,21,32)
4070 CALL HCHAR(VER,22,68)
4080 CALL HCHAR(VER,24,45)
4090 CALL HCHAR(VER,25,58)
4100 CALL HCHAR(VER,26,45,2)
4110 CALL HCHAR(VER,28,46)
4120 CALL HCHAR(VER,29,45,2)
4130 GOTO 3390
4140 FOR I=1 TO 120 STEP 3
4150 READ X,SM(I)
4160 SM(I+1)=SM(I)
4170 SM(I+2)=SM(I)
4180 ON X GOSUB 2150,2190,22
30,2270,2310,2350,2390,2430,
2470,2510
4190 NEXT I
4200 RETURN
4210 FOR J=1 TO LEN(A$)
4220 CALL HCHAR(VER,HOR+J,AS
C(SEG$(A$,J,1)))
4230 NEXT J
4240 RETURN
```

Grenze

für den TI-99/4A

Versetzen Sie sich in die Lage eines Flüchtlings, der den Versuch unternimmt, die Grenzsicherungsanlagen einer Diktatur des Jahres 2000 zu durchqueren.

Sie befinden sich in einem scharf bewachten Sektor des Grenzstreifens. Das Gebiet ist mit Minenfeldern bestückt und von geladenen Zäunen umgeben. Darin halten sich zehn Sicherheitsroboter auf, die versuchen werden, Sie auszuschalten.

Sie sind waffenlos und müssen den Automaten ausweichen, welche Ihnen hartnäckig folgen. Ihre Überlebenschance besteht darin, die Roboter in die Minenfelder zu leiten. Sie haben den Vorteil, minenverseuchtes Gelände zu erkennen. Die Roboter sehen die Gefahr nicht. Mit anderen Worten: die Minen sind günstig für Sie, solange Sie sich nur vorsichtig bewegen.

Dafür bekommen die anderen nach einiger Zeit Verstärkung, die Ihnen zu schaffen machen wird:

Die alarmierte Überwachungszentrale schickt Springroboter, die von außen in die Sperrzone eindringen und nun gemeinsam mit den anderen Maschinen versuchen, Sie einzukesseln und zu vernichten.

Vier Spielzüge später werden Selbstschußautomaten eingeschaltet, die am linken Zaun angebracht sind und 100 Stahlwürfel auf Sie feuern, sobald Sie auf der Höhe der Schußtrichter erscheinen.

Wenn Sie Pech haben, sind Sie jetzt zwischen Hochspannungszaun, Minenfeldern und Schußlinien festgenagelt, wenigstens aber sehr eingeschränkt in Ihrer Bewegungsfreiheit.

Mit einiger Geschicklichkeit schaffen Sie es aber, die Wirkung der mörderischen Kampfmaschinen gegeneinander zu richten:

- Wenn 2 Roboter das gleiche Feld besetzen wollen, so vernichtet der zuerst angekommene den nachfolgenden.
- Springroboter können Sie zu einem Satz über sich hinweg verleiten, in ein Minenfeld oder in den Hochspannungszaun, wo sie verdampfen. Wenn sich die Gelegenheit ergibt, dann lassen Sie sie in einen der Todesautomaten rennen, welcher dabei auch zerstört wird.
- Versuchen Sie, einen der Roboter zwischen sich und eine der Selbstschußanlagen zu bringen (Achtung: Der Roboter muß die Schußlinie zuerst erreichen!). Dann wird der Roboter von dem Ihnen zugedachten Schuß getroffen, der Schußtrichter ist entleert und die Minenfelder entlang der Schußrichtung sind durch Streugeschosse entschärft. Ohne diesen Trick werden Sie in der höchsten Spielstärke kaum gewinnen können.

Wenn Sie es wirklich schaffen, alle Roboter auszuschalten, dann können Sie den Zaun kurzschließen und das Gebiet verlassen.

Geben Sie Ihre Spielzüge durch eine Ziffer (1-9) entsprechend dem linksstehenden Schema ein, welches Sie auch auf Ihrem Tastentelefon wiederfinden.

'5' bedeutet "Stehenbleiben"

In verzweifelter Lage können Sie eine '0' eintippen, welche einen Sprung bewirkt, der aber leider ganz zufällig gerichtet ist und Sie retten, aber auch in die Hochspannung oder die Greifklauen eines Roboter führen kann.

Wie Ihre Gegner sich bewegen, müssen Sie herausfinden. Sie tun nichts zufällig, sondern folgen stumpfsinnig Ihrem eingebauten Programm, welches Sie schnell durchschauen werden.

Erläuterungen zum Programmlisting:

Das Programm ist in kurze logische Blöcke unterteilt, die durch REM-Zeilen voneinander getrennt sind. Der Ablauf ist so gut zu verfolgen. Eine ausführliche Kommentierung erfolgt programmintern, so daß an dieser Stelle nur noch wenige Zeilen erläutert werden sollen, die spezielle Programmiertricks enthalten, welche Anfängern erfahrungsgemäß Schwierigkeiten bereiten.

Zeile 140: Die umständlich wirkende Eingabekontrolle prüft in einer Zeile, ob ein "J" oder "j" eingegeben wurde. Andernfalls wird zu (165) verzweigt. Durch Addition des Leerstellenstrings zu QS wird es möglich, einfach ENTER einzutippen, ohne daß eine Fehlermeldung erfolgt.

Zeile 180: Verwendet die Eigenschaft, daß der Computer Aussagen Wahrheitswerte zuordnet (s. Handbuch S. 56). JOY erhält den Wert -1, wenn K=74 und 0, falls K ungleich 74 ist. Diese Technik spart umständliche IF-THEN-ELSE-Verzweigungen.

Zeile 515: Das Minuszeichen bewirkt das gleiche, als wenn an seiner Stelle "< >" stehen würde, wegen der Äquivalenz der Aussagen NR< > 32 und NR-32< > 0. (S. hierzu Handbuch S. 59)

Zeile 530: Druckt das Zeichen Nr. 91, wenn I=0 ist und sonst das Zeichen Nr. 47

Zeile 620, 635 u.a.: Die Bedingung IF Variable ... bedeutet IF Variable < > 0

660: Zeigt, wie man mit der hier erläuterten Methode (s. Kommentar zu den bisherigen Zeilen) ganze Ketten von IF-Bedingungen zu einer einzigen berechneten Verzweigung zusammen-

```
10 CALL CLEAR
15 RANDOMIZE
20 DIM X(21),Y(21)
30 DATA 46,3C5A7E2499C33C42,
47,00183C243CFFFF99,5B,00001
438447C4444,5F,000044384
4444438,60,0090440044444438

35 DATA 61,000038447844784,6
2,AA55AA55AA55AA55,91,181800
3C5A182466,92,0C10F1DE1C
244403,93,3C4299A1A199423C
40 DATA 136,000000FF,137,854
824E85A2D9222
45 FOR I=1 TO 12
50 READ K,A$
55 CALL CHAR(K,A$)
60 NEXT I
70 PRINT "G R E N Z E":::"I t
aube,rinteln":::". . . . . .
. . . . . ":::::
::::::::::
85 REM

90 REM  von DIETER TAUBE

95 REM  DINGE.STEDTWALL 16
100 REM  D-3260 RINTELN 1
105 REM
120 INPUT "Schwierigkeitsgra
d(1-3) ? ":GRAD
125 IF ABS(GRAD-2)>1 THEN 12
0
130 PRINT
135 INPUT "Kurze Erläuterung
 (J,N) ? ":Q$
140 IF ABS(ASC(Q$&" ")-90)<>
36 THEN 165
145 CALL CLEAR
150 PRINT "Der Flächtling wi
rd mit den Zahlentasten (1-9
) oder der Fernbedienung
 gesteuert":::::
155 PRINT "5 bedeutet ""Steh
enbleiben""::'0 bewirkt ein
en riskanten  Sprung"
:::"D Flüchtling bleibt steh
en,"
160 PRINT "  Spiel läuft aut
omatisch  weiter":::TAB(8
0);"-----------"::
165 PRINT :"WILLST DU EINEN
JOYSTICK  VERWENDEN ?"
170 CALL KEY(3,K,S)
175 IF S=0 THEN 170
180 JOY=(K=74)
190 REM  HAEUFIGKEIT DER MIN
ENFELDER
200 S=(11-2*GRAD)/100
205 CALL CLEAR
210 CALL SCREEN(11)
220 REM  ZEICHEN FUER ZAUN
UND SCHUSSTRICHTER

230 Z1$="101010FF1010101"
235 Z2$="00061E7E7E1E060"
240 CALL CHAR(134,Z1$)
245 CALL CHAR(135,Z1$)
250 CALL COLOR(2,9,16)
255 CALL COLOR(14,16,1)
260 CALL COLOR(13,13,4)
265 CALL COLOR(8,5,1)
295 REM  AUFBAU DES SPIELFE
LDES
325 FOR I=2 TO 21 STEP 19
330 CALL HCHAR(I,1,135,32)
335 NEXT I
340 FOR I=3 TO 31 STEP 28
345 CALL VCHAR(3,I,135,18)
350 NEXT I
355 FOR K=3 TO 20
360 FOR I=4 TO 30
365 IF RND>6 THEN 375
370 CALL HCHAR(K,I,62)
375 NEXT I
380 NEXT K
385 FOR I=2 TO 32 STEP 30
390 FOR K=4 TO 20 STEP 2
395 CALL VCHAR(K,I,62)
400 NEXT K
405 NEXT I
415 REM  SCHIESSAUTOMATEN,AN
FANGS UNSICHTBAR
425 FOR I=4 TO 20 STEP 5-GRA
D
430 CALL VCHAR(I,3,134)
435 NEXT I
465 REM  AUFSTELLUNG DER BEW
EGLICHEN FIGUREN
495 FOR I=0 TO 10
500 Z=INT(RND*18)+3
505 S=INT(RND*27)+4
510 CALL HCHAR(Z,S,NR)
515 IF NR-32 THEN 500
520 Y(I)=Z
525 X(I)=S
530 CALL HCHAR(Z,S,47-(I=0)*
44)
535 NEXT I
565 REM  ZIEHE FLUECHTLING
595 A$="Wohin gehst Du ?"
600 GOSUB 1700
605 RD3=10
610 R=0
615 R=R+1
620 IF AUTO THEN 1080
625 Z=Y(0)
630 S=X(0)
635 IF JOY THEN 1835
640 CALL KEY(3,K,ST)
645 IF ST=0 THEN 640
650 AUTO=(K=67)
655 CALL HCHAR(23,1,32,32)
660 ON 1-(K=67)-(K=53)-2*(K=
48)-3*((K<48)+(K>57))GOTO 71
5,1080,685,640,1080
665 IF K-43 THEN 715
675 REM  SPRUNG
685 X(0)=INT(RND*29)+3
690 Y(0)=INT(RND*20)+2
695 GOTO 750
705 REM  GEWAEHLTES FELD

715 IF ABS(K-53)=1 THEN 725
```

```
720 Y(0)=Y(0)+1+2*(K<52)
725 IF (K=50)+(K=56)THEN 750
730 X(0)=X(0)-1-2*(POS("253"
,STR$(K-47),1)=0)
740 REM  TEST, OB FELD BEGEH
BAR
750 CALL GCHAR(Y(0),X(0),NR)
755 IF (NR=62)+(NR>133)=0 TH
EN 895
760 IF NR>133 THEN 820
770 REM  FLUECHTLING TRITT A
UF MINE
780 I=0
785 CALL HCHAR(Z,S,32)
790 GOSUB 985
795 PRINT "Du bist auf eine
Mine":"getreten !"::
800 GOTO 1885
810 REM  SPRUNG IN DEN ZAUN
820 CALL HCHAR(Z,S,32)
825 CALL SCREEN(10)
830 FOR I=1 TO 8
835 CALL HCHAR(Y(0),X(0),91)
840 CALL SOUND(-500,-7,8)
845 CALL COLOR(8,16,4)
850 CALL HCHAR(Y(0),X(0),92)
855 CALL COLOR(8,2,4)
860 NEXT I
865 CALL SOUND(-800,-7,8)
870 PRINT "HOCHSPANNUNG !"::
"DU BIST ASCHE !"::
875 GOTO 1885
885 REM  ERLAUBTER SCHRITT
890 CALL HCHAR(Z,S,32)
900 CALL HCHAR(Y(0),X(0),91)
905 IF (NR=32)+(NR=91)THEN 1
080
915 REM  ROBOTER PACKT DICH
925 CALL COLOR(8,5,16)
930 CALL SOUND(500,-8,1)
935 FOR I=1 TO 5
940 FOR K=91 TO 47 STEP -44
945 CALL HCHAR(Y(0),X(0),K)
950 NEXT K
955 NEXT I
960 PRINT :"Du wurdest von e
inem":"frechen Computer zers:
ort !"::
965 GOTO 1885
975 REM  EXPLOSION EINER MIN
E
985 CALL HCHAR(Y(I),X(I),137
)
990 FOR K=1 TO 4
995 NEXT K
1000 CALL HCHAR(Y(I),X(I),62
)
1005 FOR K=2 TO 26 STEP 8
1010 CALL SOUND(-80,117,K,11
2,K,111,K,-6,K)
1015 NEXT K
1020 RETURN
1050 REM  BEWEGUNG DER ROBOT
ER
1080 IF SS=1 THEN 1525
1085 FOR I=1 TO 10-(R)9)*10
1090 TYP=47+(I>10)
1095 GOTO 1235
1100 NEXT I
1105 IF ROB THEN 1140
1110 PRINT "Du hast sie erle
digt !"::
1115 GOSUB 1735
1120 GOTO 1890
1130 REM  NEUE ROBOTER NACH
 DEM 8.SPIELZUG
1140 IF R=9 THEN 1165
1150 REM  SELBSTSCHUSSAUTOM
ATEN AB 13.ZUG
1160 IF R=13 THEN 1400 ELSE
615
1165 A$="Alarm wurde ausgel;
st"
1170 GOSUB 1700
1175 ROB=ROB+10
1180 FOR I=11 TO 20
1185 K=1+3*(I-10)
1190 CALL HCHAR(1,K,46)
1195 CALL SOUND(15,500,3)
1200 X(I)=K
1205 Y(I)=1
1210 NEXT I
1215 GOTO 615
1225 REM  NOCH EXISTIERENDE
ROBOTER GEHEN EINEN SCHRITT
1235 IF X(I)=0 THEN 1100
1240 Z=Y(I)
1245 S=X(I)
1250 X(I)=X(I)-(43-TYP)*SGN(
X(I)-X(0))
1255 Y(I)=Y(I)-(48-TYP)*SGN(
Y(I)-Y(0))
1260 CALL HCHAR(Z,S,32)
1265 CALL GCHAR(Y(I),X(I),NR
)
1270 CALL HCHAR(Y(I),X(I),TY
P)
1280 REM  PRUEFUNG DES BETRE
TENEN FELDES
1290 IF NR=91 THEN 925
1295 IF NR=32 THEN 1100
1300 IF NR>47 THEN 1345
1310 REM  2 ROBOTER TREFFEN
 AUFEINANDER
1320 FOR K=0 TO 24 STEP 6
1325 CALL SOUND(-30,-6,K)
1330 NEXT K
1335 CALL HCHAR(Y(I),X(I),NR
)
1340 GOTO 1400
1345 IF NR>133 THEN 1390
1355 REM  ROBOTER TRITT AUF
MINE
1365 GOSUB 985
1370 GOTO 1400
1380 REM  ROBOTER SPRINGT IN
 DEM ZAUN
1390 CALL SOUND(200,-7,8)
1395 CALL HCHAR(Y(I),X(I),13
5)
1400 ROB=ROB-1
1405 X(I)=0
1410 GOTO 1100
1440 REM  SELBSTSCHUSSANLAGE
N
1470 REM  EINSCHALTEN
1480 A$="Selbstschu-geri:te
aktiviert"
1485 GOSUB 1700
1490 CALL CHAR(134,Z2$)
1495 CALL SOUND(15,800,0,-3,
3)
1500 SS=1
1505 GOTO 615
1515 REM  SCHIESST TODESAUTO
MAT ?
1525 CALL GCHAR(Y(0),S,NN)
1530 IF NN=135 THEN 1885
1535 FOR I=4 TO X(0)
1540 CALL GCHAR(Y(0),I,N)
1545 IF (N=32)+(N=62)THEN 16
25
1550 CALL HCHAR(Y(0),4,136,I
-4)
1555 CALL SOUND(40,110,5,-3,
0)
1560 CALL HCHAR(Y(0),4,32,I-
3)
1565 CALL VCHAR(Y(0),3,135)
1570 IF N=91 THEN 1615
1580 REM  GETROFFENER ROBOTE
R WIRD MARKIERT DURCH X(K)=0
1590 ROB=ROB-1
1595 FOR K=1 TO 20
1600 IF (Y(K)-Y(0))+(X(K)-I)
THEN 1615
1605 X(K)=0
1610 GOTO 1620
1615 NEXT K
1620 IF ROB THEN 1085 ELSE 1
110
1625 NEXT I
1635 REM  ABSCHUSS DURCH SEL
BSTSCHUSSANLAGE
1645 Q$=STR$(INT(RND*20)+31-
I)
1650 CALL HCHAR(Y(0),I,92)
1655 PRINT "Treffer durch "&
Q$&' Stahl*(rfel"
1660 GOTO 1885
1675 REM  ==================

1690 REM  SCHRIFT AUF ZEILE
23
1700 FOR I=1 TO LEN(A$)
1705 CALL HCHAR(23,I+2,ASC(S
EG$(A$,I,1)))
1710 NEXT I
1715 RETURN
1725 REM  TRAELLER
1735 FOR I=1 TO 3
1740 CALL SOUND(80,168+77*I-
3*I+I,2)
1745 NEXT I
1750 CALL SOUND(300,523,2,26
2,5.659,5)
1755 RETURN
1765 REM  TRAUERAKKORDE

1775 DATA 294,349.440,247,29
4,370,220,262,330,262,311,39
2,247,294,370
1780 RESTORE 1775
1785 FOR I=1 TO 5
1790 READ A,B,C
1795 FOR K=0 TO 20 STEP 5
1800 CALL SOUND(-50,A,K,B,K,
C,K)
1805 NEXT K
1810 NEXT I
1815 RETURN
1825 REM  UNTERPROGRAMM FUE
R FERNBEDIENUNG
1835 CALL JOYST(1,H,V)
1840 IF ABS(H)+ABS(V)=0 THEN
 1860
1845 X(0)=X(0)+SGN(H)
1850 Y(0)=Y(0)-SGN(V)
1855 GOTO 750
1860 CALL KEY(3,K,ST)
1865 ON 1-(K=67)-2*(K=53)-3*
(K=48)GOTO 1835,650,1980,685
1875 REM  ABSCHLUSSROUTINE
1885 GOSUB 1775
1890 PRINT "Neuer Versuch ?"
1895 CALL KEY(3,K,S)
1900 IF S=0 THEN 1895
1905 IF K<>74 THEN 1910
1910 AUTO=0
1915 CALL SCREEN(12)
1920 CALL CHAR(134,Z1$)
1925 CALL CLEAR
1930 SS=0
1935 GOTO 265
1940 END
```

fassen kann. Im "herkömmlichen" TI-BASIC brauchte man dafür 5 Programmzeilen.

725, 755, 905 u.a.: Zeigen in besonders übersichtlicher Form die ODER-Verknüpfung (OR) zweier Aussagen im TI-BASIC

1085, 1090: Sind sehr auffällige Beispiele für die Leistungsfähigkeit der Programmoperationen mit Wahrheitswerten: Der Zähler I läuft von 1 bis 10, solange R kleiner als 10 ist. Danach, wenn R mindestens 10 ist, läuft I von 1 bis 20

Der Zeichencode "TYP" hängt vom Schleifenzähler I ab:

TYP=46, wenn I> 10, sonst ist TYP=47

Kompakter kann man die in den beiden Zeilen steckenden Fallunterscheidungen nicht programmieren.

Es wird an mehreren weiteren, hier nicht gesondert genannten, Stellen des Programms davon Gebrauch gemacht, daß sämtlichen Aussagen Zahlenwerte zugeordnet sind. Zum besseren Verständnis sei nochmals auf die (wenig beachteten) Erläuterungen des TI-BASIC-Handbuches zum LET- und IF-THEN-ELSE-Statement hingewiesen. Übrigens arbeiten andere Heim- und Personalcomputer genauso.

Nur kann es sein, daß – wie beim MICROSOFT-BASIC – wahre Aussagen nicht den Wert -1, sondern 1 erhalten.

Variablenliste:

I,K,S,ST: Lokal verwendete Zähl- und Statusvariable

A$,Q$: Zwischenspeicher für Stringdaten

GRAD,G: Gewählter Schwierigkeitsgrad

JOY: Indikator für Fernbedienung

ROB: Anzahl noch vorhandener Roboter

R: Rundenzähler (Zahl der Spielzüge)

Z,S: Zeile/Spalte des Standortes der beweglichen Figuren

NR,NN,N: ASCII-Codes von Bildschirmpositionen

AUTO: Indikator für automatische Fortsetzung des Spiels

SS: Indikator für aktivierte Selbstschußanlagen

A,B,C: Tonfrequenzen

H,V: Joystick-Rückmeldevariable

Z1$,Z2$: Hex-Strings für Zeichendefinitionen (Zaun und Schußtrichter)

Y(): Aufenthaltsorte der Spielfiguren

X(): Aufenthaltsorte der Spielfiguren

Würgman

für den TI-99/4A Grundversion

Mit "Würgman" hat uns Christian Kral aus St. Pölten in Österreich wieder ein tolles, direkt professionell geschriebenes Spiel zugeschickt. Nachdem er Sie im Mai mit "3D-Defence" (siehe CPU 5/84) in die entlegensten Winkel des Weltraums geschickt hatte, verbannt er Sie nun zur Abwechslung unter die Erde, wo Sie in einem ausweglos scheinenden Labyrinth herumliegende Ziffern verspeisen müssen. Der Name "Würgman" verdeutlicht plastisch, daß diese Leckerbissen nicht einfach zu schlucken sein werden.

Sie starten das Programm nach etwa 3-minütiger Ladezeit mit RUN, worauf das Titelbild erscheint. Pfiffig ist, daß Sie nach Drücken irgendeiner Taste die Steuertasten selbst definieren können. Dann schreibt der Computer das Labyrinth auf den Bildschirm.

Im unteren Teil werden Score, Hi-Score sowie die noch zur Verfügung stehende Zeit (TIME) angezeigt.

Ihre Aufgabe ist es, wie schon erwähnt, die sich im Labyrinth befindenden Zahlen zu fressen, während der Irrgarten seine Gestalt ständig wechselt.

Eine gefutterte Ziffer wird zum Score addiert – hat man alle verfügbaren Zahlen gefressen, wird das Zehnfache der verbliebenen Spielzeit zur Punktezahl hinzugerechnet.

Am Anfang der ersten Runde besitzt man ein Guthaben von 60 Zeiteinheiten, in jeder weiteren Runde werden 10 abgezogen. Sind alle Einheiten verbraucht, schreibt Ihr Texas "Game over" auf den Bildschirm, worauf wieder durch Drücken einer beliebigen Taste das Spiel gestartet werden kann.

```
100 REM  (C) CHRISTIAN KRAL
         WUERGMAN
110 CALL CLEAR
170 CALL SCREEN(16)
130 RANDOMIZE
140 DEF A2(A0)=INT(RND*A0+1)
150 CALL COLOR(2,6,1)
160 CALL COLOR(9,6,1)
170 CALL COLOR(10,6,1)
180 CALL COLOR(11,13,1)
190 CALL COLOR(12,9,1)
200 CALL COLOR(13,8,1)
210 CALL COLOR(14,14,1)
220 CALL COLOR(15,12,1)
230 CALL COLOR(16,4,1)
240 REM  A-D
250 DATA 3C42427E42626200,7C
22223C32327C00,3E4040406060E3
E00,7C222222232327C00
260 REM  E-H
270 DATA 7E404078606007E00,7E
404078406060000,3C42404F62623
E00,424242 7E42626200
280 REM  I-L
290 DATA 7C1010101017C00,7E
020207064630C00,424465068646
200,404040406060 7E00
300 REM  M-P
310 DATA 42665A4242626200,42
62524A4462620E,7E4242424627
E00,7C42427C40606000
320 REM  Q-T
330 DATA 7E4242426067E01,7C
42427C40646200,3C42403C06463
C00,7C1010101010180 0
340 REM  U-X
350 DATA 4242424262623C00,42
424242418180 0,4242424242 3C3B1
000,42241818242620 0
360 REM  Y-Z
370 DATA 4444281010101800,7E
0408102040 7E00
380 REM  CHR$(95)
390 DATA 000000/E000000000
400 REM  0-3
410 DATA 3C464A52646 3C00,08
18280818183C00,3C42040 8,0307
E00,3C42041C06463C00
420 REM  4-7
430 DATA 040C14247E0C0C00,7E
407C02064 3C00,3C42407C46463
C00,7E020408102000 00
440 REM  8-9
450 DATA 3C42423C46463C00,3C
42423E06463C00
460 REM  UNDERTEILE
470 REM  '-C
```

```
480 DATA 0000F1010232424,00
00F0808C42424,2424232010 0F0
000,2424C404 0F00000
490 REM  d-f
500 DATA 2424FF2424FF2424,00
00FF0000FF0000,2424242424242
424
510 REM  h-k
520 DATA 0000FF0000FF0000,00
00FF0000FF0000,0000FF0000FF0
000,0000FF0000FF0000
530 REM  l-o
540 DATA 2424242424242424,24
2424242424242424,2424242424242
424,2424242424247424
550 REM  CHR$(40)-CHR$(43)
560 DATA 2424272020272424,24
23E00404C42424,2424E7000FFF0
000,0000FF0000E7C424
570 REM  FIGUR
580 REM  p-s
590 DATA 003C3EF70FFF7E3C,00
3C7AEFF0FF7E3C,286C=AEEF6FE7
C30,387CFEF6EEEF6C38
600 REM  DREIECKE
610 REM  t-u
620 REM  (-)
630 REM  CHR$(128)-CHR$(129)

640 DATA 4060707870604000,02
060E120E060200
650 REM  DATEN LESEN
660 FOR A0=65 TO 90
670 READ A0$
680 CALL CHAR(A0,A0$)
690 NEXT A0
700 READ A0$
710 CALL CHAR(95,A0$)
720 FOR A0=48 TO 57
730 READ A0$
740 CALL CHAR(A0,A0$)
750 NEXT A0
760 FOR A0=96 TO 102
770 READ A0$
780 CALL CHAR(A0,A0$)
790 NEXT A0
800 FOR A0=104 TO 111
810 READ A0$
820 CALL CHAR(A0,A0$)
830 NEXT A0
840 FOR A0=40 TO 43
850 READ A0$
860 CALL CHAR(A0,A0$)
870 NEXT A0
880 FOR A0=112 TO 115
```

```
890 READ A0$
900 CALL CHAR(A0,A0$)
910 NEXT A0
920 FOR A0=116 TO 117
930 READ A0$
940 CALL CHAR(A0,A0$)
950 CALL CHAR(A0+7,A0$)
960 CALL CHAR(A0+12,A0$)
970 NEXT A0
980 CALL CHAR(130,"0000C6274
989E600")
990 CALL CHAR(132,"0000E6294
9898600")
1000 CALL CHAR(144,"0000C689
E9A9E600")
1010 CALL CHAR(152,"0000E589
C929C600")
1020 CALL CHAR(94,"7E7E7E7E7
E7F7E00")
1030 CALL CHAR(91,"7E1040406
0607E00")
1040 CALL CHAR(92,"0102040B1
020400 0")
1050 CALL CHAR(93,"7E020202 0
6067E00")
1060 PRINT "      'a    'eeeee0
          ff    ff'af'e
eeeeee    ff    ffb
ee'deef
cff  ff   ff"
1070 PRINT "      ff f ff   f10
+cf9eef   ff + ff   ff
f + ff    fbe*ec0e
orf f 20e2f"
1080 PRINT "      f
f     +       f'a   'a'eeee
+     +       ffba'cff
10    ff
   ffba  ff"
1090 PRINT "      ff bc ff'ea
f ba ff           ff    ffbcf
f  bcff           f-    ff
   ff   bcf"
1100 PRINT "      bbeeeeccccee
eeeeeec'ff
1110 PRINT "READY PRESS ANY
KEY TO BEGIN"
1120 LET C8=60.1
1130 LET A8=0
1140 FOR A0=1 TO 3
1150 FOR A1=0 TO 4 STEP 2
1160 CALL SOUND(-100,523,A1)
1170 NEXT A1
1180 NEXT A0
1190 FOR A0=0 TO 4 STEP 2
1200 CALL SOUND(-100,659,A0)
1210 NEXT A0
```

```
1270 FOR A0=0 TO 4 STEP 2
1220 CALL SOUND(-100,698,A0)
1240 NEXT A0
1250 FOR A0=0 TO 4 STEP 2
1260 CALL SOUND(-100,659,A0)
1270 NEXT A0
1280 FOR A2=0 TO 28 STEP 7
1290 CALL SOUND(-100,523,A2)
1300 NEXT A0
1310 CALL KEY(0,A0,A1)
1320 IF A1 THEN 4820 ELSE 13
10
1330 FOR A0=1 TO 100
1340 NEXT A0
1350 CALL SOUND(200,370,0,55
4,2)
1360 CALL SOUND(200,466,0,74
0,0,622,2)
1370 CALL SOUND(200,277,0,74
0,2)
1380 CALL SOUND(200,466,0,74
0,0,831,2)
1390 CALL SOUND(200,370,0,93
2,0)
1400 CALL SOUND(200,466,0,74
0,0,1109,2)
1410 CALL SOUND(200,277,3,12
45,2)
1420 CALL SOUND(200,466,0,74
0,0,1480,2)
1430 CALL SOUND(200,311,0,16
61,2)
1440 CALL SOUND(200,466,0,74
0,0,1864,2)
1450 CALL SOUND(200,370,0,16
61,2)
1460 CALL SOUND(200,466,0,74
0,0,1480,2)
1470 CALL SOUND(200,311,0,12
45,2)
1480 CALL SOUND(200,466,0,74
0,0,1109,2)
1490 CALL SOUND(200,370,0,93
2,2)
1500 CALL SOUND(200,466,0,74
0,0,831,2)
1510 CALL SOUND(200,277,0,74
0,2)
1520 CALL SOUND(200,466,0,74
0,0,740,2)
1530 CALL SOUND(200,208,0,74
0,2)
1540 CALL SOUND(200,466,0,74
0,0,622,2)
1550 CALL SOUND(200,277,0,55
4,2)
1560 CALL SOUND(200,466,0,74
0,0,554,2)
1570 CALL SOUND(200,208,0,55
4,2)
1580 CALL SOUND(200,466,0,74
0,0,554,2)
1590 CALL SOUND(200,554,0,11
09,2)
1600 CALL SOUND(200,587,0,11
75,2)
1610 CALL SOUND(200,622,0,12
45,2)
1620 CALL SOUND(200,698,0,13
97,2)
1630 FOR A0=1 TO 3
1640 CALL SOUND(200,740,0,14
80,2)
1650 NEXT A0
1660 CALL CLEAR
1670 FOR A0=104 TO 107
1680 CALL CHAR(A0,"0000FF000
0FF0000")
1690 CALL CHAR(A0+4,"2424242
4242424 24")
1700 NEXT A0
1710 PRINT " e+e+e+e+e+e+e+
e+e+e+e+e+e+ea f l o m m l o n
m o l l o f (idldldhd
jdhdidjdkdidjdh)"
1720 PRINT " f m a o l o n m
o l l m m l (hd,dhdidjdhdid
jdhdidkdjdh) f m o m l m
o n a l o o n f"
1730 PRINT " (idjidhdc,didkd
jdhdidjdi) f c l m m l o n
```

```
E1 ERLAEUTERTM STANDART
2460 REM
2470 CALL KEY(3,B1,B2)
2480 LET A3=A3-0.1
2490 IF INT(A3)<>A3 THEN 253
0
2500 CALL HCHAR(22,26,INT(A3
/10)+48)
2510 CALL HCHAR(22,29,(A3/10
-INT(A3/10))*10+48)
2520 IF A3=0 THEN 1490
2530 IF (B1=93)+(B1=84)+(B1=
85)+(B1=86)=0 THEN 3190
2540 LET B8=A0
2550 LET B9=A1
2560 CALL SOUND(150,-8,0)
2570 IF B1<>83 THEN 2730
2580 LET A5=114
2590 LET A0=A0-1
2600 CALL GCHAR(A0,A1,A5)
2610 GOSUB 3390
2620 IF A5=13 THEN 3190
2630 IF A5=32 THEN 2690
2640 IF ((A5=100)+(A5=101)+(
A5=42)) THEN 2660
2650 IF A7(A5-103)=2 THEN 26
90
2660 LET A0=B8
2670 LET A1=B9
2680 GOTO 2710
2690 CALL HCHAR(B8,B9,C5)
2700 CALL GCHAR(A0,A1,C5)
2710 CALL HCHAR(A0,A1,A5)
2720 GOTO 3190
2730 IF B1<>84 THEN 2890
2740 LET A5=115
2750 LET A0=A0+1
2760 CALL GCHAR(A0,A1,A5)
2770 GOSUB 3390
2780 IF A5=13 THEN 3190
2790 IF A5=32 THEN 2850
2800 IF ((A5=100)+(A5=101)+(
A5=42)) THEN 2820
2810 IF A7(A5-103)=2 THEN 28
50
2820 LET A0=B8
2830 LET A1=B9
2840 GOTO 2870
2850 CALL HCHAR(B8,B9,C5)
2860 CALL GCHAR(A0,A1,C5)
2870 CALL HCHAR(A0,A1,A5)
2880 GOTO 3190
2890 IF B1<>85 THEN 3050
2900 LET A5=112
2910 LET A1=A1-1
2920 CALL GCHAR(A0,A1,A5)
2930 GOSUB 3390
2940 IF A5=13 THEN 3190
2950 IF A5=32 THEN 3010
2960 IF ((A5=100)+(A5=102)+(
A5=40)) THEN 2980
2970 IF A7(A5-103)=2 THEN 30
10
2980 LET A0=B8
2990 LET A1=B9
3000 GOTO 3030
3010 CALL HCHAR(B8,B9,C5)
3020 CALL GCHAR(A0,A1,C5)
3030 CALL HCHAR(A0,A1,A5)
3040 GOTO 3190
3050 LET A5=113
3060 LET A1=A1+1
3070 CALL GCHAR(A0,A1,A5)
3080 IF A5=32 THEN 3160
3090 GOSUB 3390
3100 IF A5=13 THEN 3190
3110 IF ((A5=100)+(A5=102)+(
A5=41)) THEN 3130
3120 IF A7(A5-103)=2 THEN 31
00
3130 LET A0=B8
3140 LET A1=B9
3150 GOTO 3180
3160 CALL HCHAR(B8,B9,C5)
3170 CALL GCHAR(A0,A1,C5)
3180 CALL HCHAR(A0,A1,A5)
3190 REM
3200 REM  ZAEHLERTIMAUFRUFE
R ZUFALLSGENERATORE AENDERN
UND ZUGEHOERIGE VARIABLE
N SETZEN
```

```
n l o m m f (idkdjondic
(ckdhdjdidkdhd)"
1740 PRINT " f l m m l o l m
n l o m m f (idldkdhdjdidio
kdhdjdidjdh) f o l m m
n m l o l m m f"
1750 PRINT " (idkdjdhdkdidjd
hdjdkdidjdi) f a l o m m l m
n o l m m f (hdjdidkdjd
kdidhdjdidhdjdi)"
1760 PRINT " f m l o m m l o
m o l m m f (idldidkdidjdd
kdjdkdjdidk) f m o l m m
o l m m o l m f"
1770 PRINT " b*e*e*e*e*e*e*e*
e*e*e*e*e*e*e*e*e*e*e*e*e*e*
e*e*e*e*e*e*e*e*e*e*e*e*e*e*
e*e*e*e*e*e*e
```

Ich habe schon einige Computerzeitschriften gelesen. Trotzdem muß ich immer wieder zu CPU und HC zurückgreifen, da Sie immer ein Programmlisting parat haben. Nun frage ich mich, wie lange es Listings für meinen Computer (ZX Spectrum) gibt, da immer wieder neue Computer hinzukommen.
Das wäre auch schon meine einzige Sorge, ansonsten bin ich mit Ihrer Zeitschrift zufrieden.
PS.: Ich habe Ihnen ein Programmlisting beigefügt, das Sie vielleicht unter Tips und Tricks zuordnen (und natürlich veröffentlichen) können.

A. Lederer

```
  10 REM SOUND EXPERIMENTE
 100 FOR n=0 TO 29: READ z: POKE
 32030+n,z: NEXT n
 200 DATA 6,1,197,33,15,0,17,0,0
,229,205,181,3,225,17,100,0,167,
237,90,125,254,255,32,237,193,16
,230,201,0
 300 RANDOMIZE USR 32030
 400 PRINT AT 21,0;"ENDE"
 500 PAUSE 0
 600 CLS
1000 GO TO 300
```

```
   5 REM  Die DATA -Zeilen:
  10 REM 6,x,197,33,15,y,17,y,y,
229,205,181,3,225,17,y,y,167,237
,90,125,254,255,32,237,193,16,23
0,201,0
  20 REM Aendern Sie die Vari=
         ablen(x=Wiederholungen)
         (y=Dehnungen). Aber
         Vorsicht! Zum Probieren
         immer Ue 1er Faktor er=
         hoehen !
```

Antwort:
Selbstverständlich werden sowohl in CPU als auch in HOMECOMPUTER für Ihren Spectrum weiterhin Programme erscheinen. Neue Modelle kommen ja ständig auf den Markt. Wir wären nicht CPU, wenn wir dann unsere Besitzer älterer Geräte im Stich ließen!

Ich habe folgende Fragen:
1. Ich besitze seit einigen Monaten einen VC 20 und eine Datasette. Der Ladevorgang eines 16K-Bit-Programmes dauert einige Minuten. Gibt es eine Möglichkeit, den Ladevorgang zu beschleunigen ohne einen Eingriff in das Gerät vornehmen zu müssen?
2. Gibt es einen List-Schutz, der während des Ladevorgangs bereits aktiv wird?

N. Bollinger

Antwort:
Zu Ihrer ersten Frage können wir Ihnen raten, Ihren Fachhändler nach dem "Turbo-Tape"-Programm zu fragen, das ein spezielles TOS, also ein Tape Operating System darstellt und eine wesentliche Verkürzung der Ladezeit durch Erhöhung der Baudrate bietet.
Was den von Ihnen gewünschten List-Schutz betrifft, so gibt es ihn in Deutschland noch nicht. Entwickelt ist er aber bereits und wird mit Erfolg in vielen Industrieprogrammen benutzt.

Ich habe schon einige Computerzeitschriften gelesen. Trotzdem muß ich immer wieder zu CPU und HC zurückgreifen, da Sie immer ein Programmlisting parat haben. Nun frage ich mich, wie lange es Listings für meinen Computer (ZX Spectrum) gibt, da immer wieder neue Computer hinzu kommen.
Das wäre auch schon meine einzige Sorge, ansonsten bin ich mit Ihrer Zeitschrift zufrieden.
PS.: Ich habe Ihnen ein Programmlisting beigefügt, das Sie vielleicht unter Tips und Tricks zuordnen (und natürlich veröffentlichen) können.

A. Lederer

Antwort:
Selbstverständlich werden sowohl in CPU als auch in HOMECOMPUTER für Ihren Spectrum weiterhin Programme erscheinen. Neue Modelle kommen ja ständig auf den Markt. Wir wären nicht CPU, wenn wir dann unsere Besitzer älterer Geräte im Stich ließen!

Zum Brief von M. Kötfing, aus einem vorhergehenden HC-Heft, möchte ich folgendes anführen: Mir ist dieser Fehler auch schon aufgefallen. In meinem Fall kam es zu dem Fehler, als ich das Modul, ohne auf das Titelbild zurückzuschalten, in die Konsole einschob. Dabei kann es passieren, daß der Resetbefehl nicht ausgeführt wird. Um solche Fehler zu vermeiden, schalte ich meinen TI erst aus, schiebe das Modul in den Slot und schalte dann erst wieder ein. So steht es auch in einigen Betriebsanleitungen zu den Modulen. Es gibt aber auch Module, welche nicht auf allen TI 99/4A laufen. Es kommt auf das Copyright-Jahr an.
Zur Speicherplatzermittlung von F. Feuster und W. Schleiden kann man nur schmunzeln. Soweit mir bekannt ist, setzt sich 1 Byte aus 8 Bit zusammen und nicht aus 8.50115 Bit. Richtig wäre:
1 RAM=RAM+8
2 GOSUB 1
RUN
MEMORY FULL in 1
Print RAM ergibt im Ext. Basic 13.920, im TI Basic ergibt das 14.536 Byte.

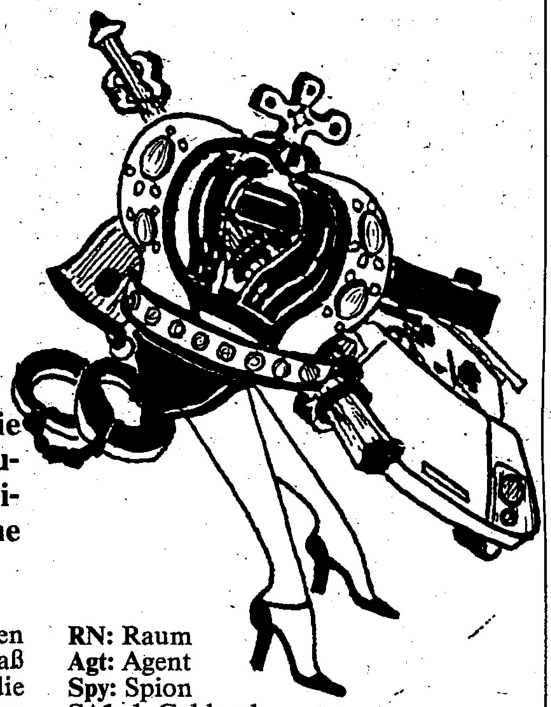
W. Bürger

Ich würde mich freuen, wenn Sie mir folgende Fragen beantworten könnten:
Ich besitze Listings für den Spectrum, in denen Buchstaben unterstrichen sind. Wie erreiche ich das? Dann habe ich noch eine Frage zum Computerposter "6+7": Im Vorprogramm heißt es in einer REM-Zeile "Gestartet wird mit RAND USR 16768". Was ist damit gemeint? Und zu guter letzt: Woher bekomme ich eine deutsche Anleitung zu Fighter Pilot?

U. Hermanns

Antwort:
Zu Ihrer ersten Frage können wir Ihnen mitteilen, daß es sich hier um ein Graphikzeichen handelt, d.h. eine Kenntlichmachung, daß Sie vor Betätigung der jeweiligen Zeichen (-Tasten) in den Graphic Mode gehen müssen.
Frage Nr. 2: Hier handelt es sich um eine Verwechslung zwischen Spectrum und ZX-81, für was wir um Entschuldigung bitten.
Die letzte Frage müssen wir unbeantwortet an die Generalvertretung von Sinclair weitergeben. Schreiben Sie an SINCLAIR DEUTSCHLAND GmbH, Postfach, 8012 Ottobrunn

The Queens Agent
für den TI-99/4A



Der Königin von England wurden von einem feindlichen Agenten die Kronjuwelen gestohlen. Ihre Aufgabe ist es, den Agenten auszuschalten und die Juwelen zurückzubringen. Er versteckt sich in einem Haus mit 15 Räumen, in dem sich außerdem ein Spion und eine Bombe befinden.

Zum Spiel:
Es werden angezeigt:
Anzahl der restlichen Schritte
Anzahl der noch im Haus befindlichen Juwelensäcke
Anzahl der noch zur Verfügung stehenden Schüsse
Score
Anzeige der Raumnummer
In dem mit + gekennzeichneten Feld können folgende Anzeigen auftauchen: Agent near, Spy near, Bomb near, Spy dead, Jewels found, Computer set room
In der linken Bildschirmhälfte wird in einer Grafik der jeweilige Raum dargestellt.
Zu Beginn erscheint das Titelbild mit der Melodie von Big Ben. Die Melodie wiederholt sich so oft, bis das Spiel durch Drücken einer Taste gestartet wird.
Sie befinden sich nun im Raum 1. Der Computer nimmt Befehle an, sobald oben rechts das "C" und der Pfeil im Wahlfeld: go? Shoot? erscheint.
Sie haben nun die Wahl zwischen: gehen- und schießen.
Nach Drücken der entsprechenden Taste ertönt ein Warnsignal, das darauf hinweist, daß die Richtung noch nicht feststeht. Nach Eingabe der Richtung wird der Befehl ausgeführt und Sie wechseln bzw. schießen in den gewählten Raum.
Benutzen Sie einen der Ausgänge, wählt der Computer einen Raum aus und setzt Sie dort ab. Dabei werden Ihnen allerdings 10 Schritte abgezogen und es besteht die Gefahr, in den Räumen mit dem Agenten, dem Spion oder der Bombe zu landen, wodurch das Spiel sofort beendet wäre.
Eine Anzeige, wie z.B. Agent near, bedeutet, daß sich das angezeigte entweder in einem Raum über, unter, links oder rechts von Ihnen befindet. Die dort angezeigten Dinge verändern ihre Position nicht, sodaß Sie mit der richtigen Strategie herausfinden, in welchem Raum sich was befindet. Töten Sie den Agenten, ist das Spiel beendet, sodaß Sie keine Gelegenheit mehr haben, die Juwelen zu finden. Man sollte sich also merken, wo der Agent ist und ihn erst zum Schluß ausschalten.
Sind alle Schritte verbraucht, ist das Spiel ebenfalls beendet (wenn Schritte < 0).
Kommt man mit dem letzten Befehl mehr als einen Schritt unter Null, wird der Endscore als "C"-Score angezeigt, was einen Minuswert bedeutet.
Bei der Endabrechnung wird die Restschrittzahl mitbewertet, also nicht so oft die Ausgänge benutzen!!
Ein Schuß kostet Sie 3 Schritte, für einen gefunden Juwelensack erhalten Sie 5 Schritte. Der Raum mit der Bombe bleibt für Sie gesperrt. Betreten Sie ihn trotzdem, ist das Spiel zu Ende.
Töten Sie den Spion, wird dies auf dem Display und akustisch angezeigt. Das Spiel wird dadurch aber nicht beendet.
Da Ihnen nur 2 Schüsse zur Verfügung stehen und Sie den Agenten und den Spion nur dadurch ausschalten können, überlegen Sie erst genau, wo Sie sich befinden. Diese Räume dürfen nicht betreten werden. Bei dem Versuch ist das Spiel beendet.
Das Programm weist unerlaubte Schritte zurück: z.B: Schritt durch eine Wand, Schuß von Raum 10 nach 11 etc.
Sicher wird es einige Zeit dauern, bis Sie das Spiel im Griff haben, aber es macht immermehr Spaß!!!
Noch ein Tip:
Angenommen, Sie befinden sich in Raum 10 und es wird: Spy near angezeigt, so kann sich der Spion in Raum 5, 9, 15 aber auch in Raum 11 befinden!!!!!

Variablen: **V:** Anzahl der Schritte
SC: Punktezähler
B: Bonusberechnung
BO: Bombe
RN: Raum
Agt: Agent
Spy: Spion
SA1: 1. Geldsack
SA2: 2. Geldsack
SA3: 3. Geldsack
DD: Zähler (Geldsäcke)
a 1: Position Schriftanzeige
A 2: Position Schriftanzeige
A 3: Position Schriftanzeige
A 4: Position Schriftanzeige
A 5: Position Schriftanzeige
SH: Zähler Schüsse

Programmbeschreibung:
110-400 Zeichendefinierung Spielfeld
410-570: Farbfestlegung Spielfeld
580-790: Datas für ersten Raum-, Spielfeldaufbau
800-890: Datas für Schrift erstes Bild
900-940: Variablen
950-1230: Auswahl der Räume für die Spielfiguren
1330-1640: Überprüfung der Nebenräume
1660-1710: Überprüfung, ob der Raum besetzt ist
1990-2210: Überprüfung der Schritte auf Gültigkeit
2300-2780: Raumauswahl des Computer
2790-3200: Datas zum Aufbau der verschiedenen Räume
3210-4010: Schriftprogramm Spielablauf
4030-4910: Bewertung für Schüsse, Juwelen, Schrittabrechnung
4920-4990: Zeichendefinierung Titelbild
5000-5080: Farbfestlegung Titelbild
5090-5300: Aufbau Titelbild
5310-5520: Sound (Titelbild/Scoretabelle)
5630-5770: Scoretabelle (Endabrechnung)
5780-5840: Abfrage, ob noch ein Spiel
5890-6060: Schriftprogramm, wie weit die Mission erfüllt ist

```
100 GOSUB 4920
110 CALL CLEAR
120 CALL CHAR(42,"FF7F3F1F0F
070301")
130 CALL CHAR(43,"FFFFFFFFFF
FFFFFF")
140 CALL CHAR(96,"0103070F1F
3F7FFF")
150 CALL CHAR(97,"FFFEFCF8F0
E0C080")
160 CALL CHAR(98,"FFFFFFFFFF
FFFFFF")
170 CALL CHAR(104,"FFFFFFFFF
FFFFFFF")
180 CALL CHAR(105,"80C0E0F0F
8FCFEFF")
190 CALL CHAR(112,"80C0E0F0F
8FCFEFF")
200 CALL CHAR(113,"FFFFFFFFF
FFFFFFF")
210 CALL CHAR(114,"FFFEFCF8F
0E0C080")
220 CALL CHAR(120,"0103070F1
F3F7FFF")
230 CALL CHAR(128,"8080808080
808080")
240 CALL CHAR(129,"FFFFFFFFF
FFFFFFF")
250 CALL CHAR(130,"000000000
00000FF")
260 CALL CHAR(131,"FF")
270 CALL CHAR(136,"010101010
1010101")
280 CALL CHAR(137,"80C0E0F0F
8FCFEFF")
290 CALL CHAR(138,"0103070F1
F3F7FFF")
300 CALL CHAR(139,"007E46400
40E0404")
310 CALL CHAR(151,"0103070F1
F3F7FFF")
320 CALL CHAR(59,"3C4299A1A1
99423C")
330 CALL CHAR(152,"000000001
0307FFF")
340 CALL CHAR(153,"7F301")
350 CALL CHAR(154,"000000000
00000FF")
360 CALL CHAR(155,"010101010
1010101")
370 CALL CHAR(156,"FF")
380 CALL CHAR(157,"8080808080
808080")
390 CALL CHAR(158,"006B3E1C3
66F773E")
400 CALL CHAR(159,"818181818
1818181")
410 CALL SCREEN(8)
420 CALL COLOR(1,5,5)
430 CALL COLOR(2,11,15)
440 CALL COLOR(3,16,5)
450 CALL COLOR(4,16,5)
460 CALL COLOR(5,16,5)
470 CALL COLOR(6,16,5)
480 CALL COLOR(7,16,5)
490 CALL COLOR(8,16,5)
500 CALL COLOR(9,2,11)
510 CALL COLOR(10,11,2)
520 CALL COLOR(11,16,1)
530 CALL COLOR(12,11,8)
540 CALL COLOR(13,15,1)
550 CALL COLOR(14,15,16)
560 CALL COLOR(15,15,11)
570 CALL COLOR(16,2,5)
580 DATA 1,3,154,16,19,3,156
,16,2,3,105,1,2,4,98,14,2,18
,97,1,3,4,105,1
590 DATA 3,5,98,12,3,17,97,1
,4,5,105,1,4,6,98,10,4,16,97
,1
600 DATA 5,6,105,1,5,7,98,8,
5,15,97,1,6,7,105,1,6,8,98,6
,6,14,97,1
610 DATA 7,8,105,1,7,9,98,4,
7,13,97,1,13,8,151,1,13,9,12
9,4,13,13,42,1
620 DATA 14,7,151,1,14,8,129
,6,14,14,42,1,15,6,138,1,15,
7,129,8,15,15,42,1
630 DATA 16,5,138,1,16,6,129
,10,16,16,42,1,17,4,151,1,17
,5,129,12,17,17,42,1
640 DATA 18,3,151,1,18,4,129
,14,18,18,42,1,11,20,154,8,1
4,20,156,8
650 DATA 15,20,154,8,18,20,1
56,8
660 RESTORE 580
670 FOR T=1 TO 42
680 READ A,B,C,D
690 CALL HCHAR(A,B,C,D)
700 NEXT T
710 DATA 2,2,155,17,2,19,157
,17,9,10,113,4,9,11,113,4,9,
5,113,7,9,6,113,6
720 DATA 3,3,104,15,4,4,104,
13,5,5,104,4,6,6,104,3,7,7,1
04,7,8,8,104,5
730 DATA 8,13,43,5,7,14,43,7
,6,15,43,9,5,16,43,11,4,17,4
3,13
740 DATA 3,18,43,15,12,19,15
9,2,12,28,157,2,16,19,159,2,
16,28,157,2,12,6,139,1,1
1,11,139,1
750 RESTORE 710
760 FOR T=1 TO 24
770 READ A,B,C,D
780 CALL VCHAR(A,B,C,D)
790 NEXT T
800 DATA 3,20,76,3,21,69,3,2
2,70,3,23,84,3,24,62,3,25,76
810 DATA 5,20,82,5,21,73,5,2
2,71,5,23,72,5,24,84,5,25,62
,5,26,82
820 DATA 7,20,79,7,21,78,7,2
2,87,7,23,65,7,24,82,7,25,68
,7,26,62,7,27,79
830 DATA 9,20,66,9,21,65,9,2
2,67,9,23,75,9,24,62,9,25,66
840 DATA 12,21,71,12,22,79,1
2,23,63,13,21,83,13,22,72,13
,23,79,13,24,79,13,25,84
,13,26,63
850 RESTORE 800
860 FOR T=1 TO 36
870 READ A,B,C
880 CALL HCHAR(A,B,C)
890 NEXT T
900 RN=1
910 DD=3
920 SH=2
930 V=15
940 SC=0
950 RANDOMIZE
960 AGT=INT(RND*15)+1
970 IF AGT=1 THEN 960
980 SPY=INT(RND*15)+1
990 IF SPY=1 THEN 980
1000 IF SPY=AGT THEN 980
1010 SA1=INT(RND*15)+1
1020 IF SA1=1 THEN 1010
1030 IF SA1=AGT THEN 1010
1040 IF SA1=SPY THEN 1010
1050 SA2=INT(RND*15)+1
1060 IF SA2=1 THEN 1050
1070 IF SA2=AGT THEN 1050
1080 IF SA2=SPY THEN 1050
1090 IF SA2=SA1 THEN 1050
1100 SA3=INT(RND*15)+1
1110 IF SA3=1 THEN 1100
1120 IF SA3=AGT THEN 1100
1130 IF SA3=SPY THEN 1100
1140 IF SA3=SA1 THEN 1100
1150 IF SA3=SA2 THEN 1100
1160 BO=INT(RND*15)+1
1170 IF BO=1 THEN 1160
1180 IF BO=AGT THEN 1160
1190 IF BO=SPY THEN 1160
1200 IF BO=SA1 THEN 1160
1210 IF BO=SA2 THEN 1160
1220 IF BO=SA3 THEN 1160
1230 GOTO 1270
1240 IF RN<6 THEN 1270
1250 IF RN<11 THEN 1290
1260 GOTO 1310
1270 GOSUB 2790
1280 GOTO 1320
1290 GOSUB 2930
1300 GOTO 1320
1310 GOSUB 3070
1320 GOSUB 2300
1330 IF RN=AGT-1 THEN 1380
1340 IF RN=AGT+1 THEN 1380
1350 IF RN=AGT-5 THEN 1380
1360 IF RN=AGT+5 THEN 1380
1370 GOTO 1440
1380 FOR A=1 TO 3
1390 CALL SOUND(250,-1,0)
1400 FOR T=1 TO 150
1410 NEXT T
1420 NEXT A
1430 GOSUB 3390
1440 IF RN=SPY-1 THEN 1490
1450 IF RN=SPY+1 THEN 1490
1460 IF RN=SPY+5 THEN 1490
1470 IF RN=SPY-5 THEN 1490
1480 GOTO 1550
1490 FOR A=1 TO 3
1500 CALL SOUND(250,-1,0)
1510 FOR T=1 TO 150
1520 NEXT T
1530 NEXT A
1540 GOSUB 3480
1550 IF RN=BO-1 THEN 1600
1560 IF RN=BO+1 THEN 1600
1570 IF RN=BO+5 THEN 1600
1580 IF RN=BO-5 THEN 1600
1590 GOTO 1660
1600 FOR A=1 TO 3
1610 CALL SOUND(250,-1,0)
1620 FOR T=1 TO 150
1630 NEXT T
1640 NEXT A
1650 GOSUB 3660
1660 IF RN=SA1 THEN 1740
1670 IF RN=SA2 THEN 1740
1680 IF RN=SA3 THEN 1740
1690 IF RN=SPY THEN 3920
1700 IF RN=AGT THEN 3920
1710 IF RN=BO THEN 3920
1720 GOSUB 3750
1730 GOTO 1760
1740 GOTO 4030
1750 GOSUB 3570
1760 GOSUB 3750
1770 GOTO 4220
1780 CALL KEY(0,K,S)
1790 CALL HCHAR(2,27,59)
1800 CALL SOUND(100,1760,0)
1810 IF S=0 THEN 1780
1820 CALL HCHAR(2,27,32)
1830 CALL HCHAR(12,12,32)
1840 A1=20
1850 A2=2
1860 A3=2
1870 A4=20
1880 A5=20
1890 IF K=79 THEN 1900 ELSE
1910
1900 IF BEF=1 THEN 4360 ELSE
 1980
1910 IF K=66 THEN 1920 ELSE
1930
1920 IF BEF=1 THEN 4450 ELSE
 2040
1930 IF K=76 THEN 1940 ELSE
1950
1940 IF BEF=1 THEN 4540 ELSE
 2100
1950 IF K=82 THEN 1960 ELSE
1970
1960 IF BEF=1 THEN 4610 ELSE
 2160
1970 GOTO 1780
1980 RN=RN+1
1990 IF RN=16 THEN 2220
2000 IF RN=11 THEN 2220
2010 IF RN=6 THEN 2220
2020 GOSUB 4830
2030 GOTO 1240
2040 RN=RN-1
2050 IF RN=10 THEN 2260
2060 IF RN=5 THEN 2260
2070 IF RN=0 THEN 2260
2080 GOSUB 4830
2090 GOTO 1240
2100 RN=RN+5
2110 IF RN>15 THEN 2140
2120 GOSUB 4830
2130 GOTO 1240
2140 RN=RN-5
2150 GOTO 1240
```

```
2160 RN=RN-5
2170 IF RN<=0 THEN 2200
2180 GOSUB 4830
2190 GOTO 1240
2200 RN=RN+5
2210 GOTO 1240
2220 RN=INT(RND*15)+1
2230 GOSUB 4860
2240 GOSUB 3210
2250 GOTO 1240
2260 RN=INT(RND*15)+1
2270 GOSUB 4860
2280 GOSUB 3210
2290 GOTO 1240
2300 A1=20
2310 A2=2
2320 A3=2
2330 A4=20
2340 A5=20
2350 RN$=STR$(RN)
2360 A$="ROOM NO:"&RN$
2370 GOSUB 2530
2380 SH$=STR$(SH)
2390 A$="SHOOTS:"&SH$
2400 GOSUB 2690
2410 DD$=STR$(DD)
2420 A$="SACKS :"&DD$
2430 GOSUB 2740
2440 V$=STR$(V)
2450 A$="STEPS LEFT:"&V$
2460 CALL HCHAR(20,13,32,2)
2470 GOSUB 2590
2480 SC$=STR$(SC)
2490 A$="SCORE:"&SC$
2500 GOSUB 2640
2510 CALL HCHAR(2,27,59)
2520 RETURN
2530 CALL HCHAR(22,28,32,2)
2540 FOR N2=1 TO LEN(A$)
2550 CALL HCHAR(22,A1,ASC(SE
G$(A$,N2,1)))
2560 A1=A1+1
2570 NEXT N2
2580 RETURN
2590 FOR N2=1 TO LEN(A$)
2600 CALL HCHAR(20,A2,ASC(SE
G$(A$,N2,1)))
2610 A2=A2+1
2620 NEXT N2
2630 RETURN
2640 FOR N2=1 TO LEN(A$)
2650 CALL HCHAR(22,A3,ASC(SE
G$(A$,N2,1)))
2660 A3=A3+1
2670 NEXT N2
2680 RETURN
2690 FOR N2=1 TO LEN(A$)
2700 CALL HCHAR(20,A4,ASC(SE
G$(A$,N2,1)))
2710 A4=A4+1
2720 NEXT N2
2730 RETURN
2740 FOR N2=1 TO LEN(A$)
2750 CALL HCHAR(19,A5,ASC(SE
G$(A$,N2,1)))
2760 A5=A5+1
2770 NEXT N2
2780 RETURN
2790 REM
2800 DATA 15,6,138,1,16,5,13
8,1,15,15,42,1,16,16,42,1
2810 RESTORE 2800
2820 FOR T=1 TO 4
2830 READ A,B,C,D
2840 CALL HCHAR(A,B,C,D)
2850 NEXT T
2860 DATA 9,5,113,7,9,6,113,
6,9,15,43,6,9,16,43,7,12,6,1
39,1
2870 RESTORE 2860
2880 FOR T=1 TO 5
2890 READ A,B,C,D
2900 CALL VCHAR(A,B,C,D)
2910 NEXT T
2920 RETURN
2930 REM  *RAUM 2*
2940 DATA 15,15,137,1,16,16,
137,1,15,6,138,1,16,5,138,1
2950 RESTORE 2940
2960 FOR T=1 TO 4
2970 READ A,B,C,D
2980 CALL HCHAR(A,B,C,D)
2990 NEXT T
3000 DATA 9,15,113,6,9,16,11
3,7,9,5,113,7,9,6,113,6,12,1
6,139,1,12,6,139,1
3010 RESTORE 3000
3020 FOR T=1 TO 6
3030 READ A,B,C,D
3040 CALL VCHAR(A,B,C,D)
3050 NEXT T
3060 RETURN
3070 REM  *RAUM 3*
3080 DATA 15,6,151,1,16,5,15
1,1,15,15,137,1,16,16,137,1
3090 RESTORE 3080
3100 FOR T=1 TO 4
3110 READ A,B,C,D
3120 CALL HCHAR(A,B,C,D)
3130 NEXT T
3140 DATA 9,5,104,7,9,6,104,
6,9,15,113,6,9,16,113,7,12,1
6,139,1
3150 RESTORE 3140
3160 FOR T=1 TO 5
3170 READ A,B,C,D
3180 CALL VCHAR(A,B,C,D)
3190 NEXT T
3200 RETURN
3210 A$="COMPUTER"
3220 Q=16
3230 W=20
3240 GOSUB 3840
3250 A$="SET ROOM"
3260 Q=17
3270 W=20
3280 GOSUB 3840
3290 RETURN
3300 A$=" SPY    "
3310 Q=16
3320 W=20
3330 GOSUB 3840
3340 A$=" DEAD   "
3350 Q=17
3360 W=20
3370 GOSUB 3840
3380 RETURN
3390 A$=" AGENT  "
3400 Q=16
3410 W=20
3420 GOSUB 3840
3430 A$=" NEAR   "
3440 Q=17
3450 W=20
3460 GOSUB 3840
3470 RETURN
3480 A$=" SPY    "
3490 Q=16
3500 W=20
3510 GOSUB 3840
3520 A$=" NEAR   "
3530 Q=17
3540 W=20
3550 GOSUB 3840
3560 RETURN
3570 A$=" JEWELS "
3580 Q=16
3590 W=20
3600 GOSUB 3840
3610 A$=" FOUND  "
3620 Q=17
3630 W=20
3640 GOSUB 3840
3650 RETURN
3660 A$=" BOMB   "
3670 Q=16
3680 W=20
3690 GOSUB 3840
3700 A$=" NEAR   "
3710 Q=17
3720 W=20
3730 GOSUB 3840
3740 RETURN
3750 A$="        "
3760 Q=16
3770 W=20
3780 GOSUB 3840
3790 A$="        "
3800 Q=17
3810 W=20
3820 GOSUB 3840
3830 RETURN
3840 FOR DRI=1 TO LEN(A$)
3850 CODE=ASC(SEG$(A$,DRI,1)
)
3860 GOSUB 3900
3870 W=W+1
3880 NEXT DRI
3890 RETURN
3900 CALL HCHAR(Q,W,CODE)
3910 RETURN
3920 SC=SC-500
3930 CALL CLEAR
3940 CALL SCREEN(5)
3950 PRINT "    YOU WERE NOT
 CAREFUL    "
3960 PRINT
3970 PRINT "    THE FOE HAS
GOT YOU    "
3980 PRINT ::::::
3990 FOR T=1 TO 2000
4000 NEXT T
4010 GOTO 5590
4020 END
4030 DD=DD-1
4040 FOR I=252 TO 524 STEP 3
6
4050 CALL SOUND(-50,I,0,-1,1
0)
4060 NEXT I
4070 V=V+5
4080 SC=SC+500
4090 CALL HCHAR(12,12,158)
4100 IF RN=SA1 THEN 4160
4110 IF RN=SA2 THEN 4180
4120 IF RN=SA3 THEN 4200
4130 GOSUB 2300
4140 GOSUB 3570
4150 GOTO 1760
4160 SA1=0
4170 GOTO 4130
4180 SA2=0
4190 GOTO 4130
4200 SA3=0
4210 GOTO 4130
4220 CALL KEY(0,KY,ST)
4230 CALL HCHAR(12,27,152)
4240 CALL HCHAR(13,27,153)
4250 IF ST=0 THEN 4220
4260 CALL HCHAR(12,27,32)
4270 CALL HCHAR(13,27,32)
4280 IF KY=83 THEN 4310
4290 IF KY=71 THEN 4340
4300 GOTO 4220
4310 BEF=1
4320 IF SH=0 THEN 4220
4330 GOTO 1780
4340 BEF=0
4350 GOTO 1780
4360 IF RN+1=16 THEN 4220
4370 IF RN+1=11 THEN 4220
4380 IF RN+1=6 THEN 4220
4390 IF RN+1=AGT THEN 4680
4400 IF RN+1=SPY THEN 4710
4410 SH=SH-1
4420 GOSUB 4890
4430 GOSUB 2300
4440 GOTO 1760
4450 IF RN-1=10 THEN 4220
4460 IF RN-1=5 THEN 4220
4470 IF RN-1=0 THEN 4220
4480 IF RN-1=AGT THEN 4680
4490 IF RN-1=SPY THEN 4710
4500 SH=SH-1
4510 GOSUB 4890
4520 GOSUB 2300
4530 GOTO 1760
4540 IF RN+5>15 THEN 4220
4550 IF RN+5=AGT THEN 4680
4560 IF RN+5=SPY THEN 4710
4570 SH=SH-1
4580 GOSUB 4890
4590 GOSUB 2300
4600 GOTO 1760
4610 IF RN-5<0 THEN 4220
4620 IF RN-5=AGT THEN 4680
4630 IF RN-5=SPY THEN 4710
4640 SH=SH-1
4650 GOSUB 4890
4660 GOSUB 2300
4670 GOTO 1760
4680 SC=SC+1500
4690 GOTO 5890
```

```
4700 END
4710 SC=SC+1000
4720 FOR I=139 TO 134 STEP
1
4730 CALL SOUND(-100,I,0)
4740 NEXT I
4750 CALL SOUND(150,-7,0)
4760 SPY=100
4770 GOSUB 4890
4780 SH=SH-1
4790 GOSUB 3300
4800 GOSUB 2300
4810 GOTO 1760
4820 GOTO 1760
4830 V=V-1
4840 IF V<0 THEN 5590
4850 RETURN
4860 V=V-10
4870 IF V<0 THEN 5590
4880 RETURN
4890 V=V-3
4900 IF V<0 THEN 5590
4910 RETURN
4920 CALL CLEAR
4930 CALL CHAR(40,"FFFFFFFFF
FFFFFFFFF")
4940 CALL CHAR(41,"FFFEFCF8F
0E0C080")
4950 CALL CHAR(42,"0103070F1
F3F7FFF")
4960 CALL CHAR(43,"80C0E0F0F
8FCFEFF")
4970 CALL CHAR(44,"FF7F3F1F0
F070301")
4980 CALL CHAR(45,"3C4299A1A
199423C")
4990 CALL CHAR(64,"FFFFFFFFF
FFFFFFF")
5000 CALL SCREEN(2)
5010 CALL COLOR(1,2,2)
5020 CALL COLOR(2,12,2)
5030 CALL COLOR(3,16,2)
5040 CALL COLOR(4,16,2)
5050 CALL COLOR(5,16,2)
5060 CALL COLOR(6,16,2)
5070 CALL COLOR(7,16,2)
5080 CALL COLOR(8,16,2)
5090 PRINT "          @@@ @*@+
@@@"
5100 PRINT "  +       @  @(@(
@          *"
5110 PRINT "  (+      @ *@@@(
@@       *("
5120 PRINT "  ((+  *+@*(@(@(
@(+**  *(("
5130 PRINT "  (((+*((@((@(@(
@@@((+*(((("
5140 PRINT "  ((((((((((((((
(((((((((("
5150 PRINT " @@@(@(@(@@@(@@@
(@@((@(@@@@"
5160 PRINT " @(@(@(@(@((@((
(@@@(@(@((
5170 PRINT " @(@(@(@(@@((@@(
(@(@@@(@@@@"
5180 PRINT " @(@(@(@(@((@((
(@((@@((((@"
5190 PRINT " @@@(@@@(@@@(@@@
(@((@(@@@@"
5200 PRINT "  ((@((((((((((
(((((((((("
5210 PRINT "  ((((((((((((((
(((((((((("
5220 PRINT "  ((@@@(@@@(@@@(
@@(@(@@@("
5230 PRINT "  ((@(@(@((@(((
@@@(@((@(("
5240 PRINT "  ((@@@(@(@(@@((
@(@@@((@(("
5250 PRINT "  ,(@(@(@(@(@((
@((@@((@()"
5260 PRINT "   ,@(@(@@@(@@@(
@((@((@)"
5270 PRINT
5280 PRINT " -1984 PROGRAMM:
J.SCHLUETER"
5290 PRINT
5300 PRINT " SOUND.: GRAFIK:
F.SCHLUETER"
5310 CALL SOUND(400,220,0,17
5,10)
5320 FOR I=1 TO 100
5330 NEXT I
5340 CALL SOUND(400,175,0,34
9,10)
5350 FOR I=1 TO 100
5360 NEXT I
5370 CALL SOUND(400,196,0,13
1,10)
5380 FOR I=1 TO 100
5390 NEXT I
5400 CALL SOUND(700,131,0,26
2,10)
5410 FOR I=1 TO 400
5420 NEXT I
5430 CALL SOUND(400,131,0,26
2,10)
5440 FOR I=1 TO 100
5450 NEXT I
5460 CALL SOUND(400,196,0,13
1,10)
5470 FOR I=1 TO 100
5480 NEXT I
5490 CALL SOUND(400,220,0,17
5,10)
5500 FOR I=1 TO 100
5510 NEXT I
5520 CALL SOUND(700,175,0,34
9,10)
5530 FOR I=1 TO 30
5540 CALL KEY(0,K,S)
5550 IF S=0 THEN 5560 ELSE 5
580
5560 NEXT I
5570 GOTO 5310
5580 RETURN
5590 REM **AUSWERTUNG**
5600 CALL CLEAR
5610 CALL SCREEN(5)
5620 IF V=-1 THEN 5630 ELSE
5640
5630 V=0
5640 SC=SC+(V*50)
5650 B=V*50
5660 PRINT "      G A M E  O
 V E R"
5670 PRINT
5680 PRINT "    S C O R E  T
 A B L E"
5690 PRINT :::
5700 PRINT "    STEPS LEFT:"
;V
5710 PRINT ::
5720 PRINT "    BONUS:";B
5730 PRINT ::
5740 PRINT "    SCORE:";SC
5750 PRINT ::
5760 PRINT "        PRESS ANY
 KEY"
5770 GOSUB 5310
5780 CALL CLEAR
5790 CALL SCREEN(5)
5800 PRINT " , A N O T H E R
  G A M E"
5810 PRINT ::
5820 PRINT "            YES<>
NO"
5830 PRINT :::::
5840 CALL KEY(0,K,S)
5850 IF S=0 THEN 5840
5860 IF K=89 THEN 110
5870 IF K=78 THEN 5880 ELSE
5840
5880 END
5890 IF SA1<>0 THEN 5950
5900 IF SA2<>0 THEN 5950
5910 IF SA3<>0 THEN 5950
5920 A$="        CONGRATULATI
ON"
5930 B$="  THE JEWELS ARE BA
CK HOME"
5935 SC=SC+750
5940 GOTO 5980
5950 A$="YOU FORGOT TO BRING
 BACK THE"
5960 B$="JEWELS  BE CAREFUL
NEXT TIME"
5970 GOTO 5980
5980 CALL CLEAR
5990 CALL SCREEN(5)
6000 PRINT A$
6010 PRINT
6020 PRINT B$
6030 PRINT :::::::
6040 FOR I=1 TO 2000
6050 NEXT I
6060 GOTO 5590
```


IMPRESSUM

Homecomputer
erscheint monatlich im
Roeske Verlag, Eschwege

Herausgeber:
Ralph Roeske

Redaktion:
Ralph Roeske (Chefredakteur, verantwortlich)
Christian Widuch, Gertrud Marx-Fischer,
Stefan Kaus

Freie Mitarbeiter:
A. Koschel, M. Hoppe, R. Ivannejad,
W. Krausler, M. Wolf, J. u. F. Schlüter,
P. Schmitz, K. Schwanz, W. Kalus, K. Ezcan,
F. Ohmann, R. Reifert

Gestaltung u. Fotografie:
G. Köberich, R. Wells

Programmierarbeiten:
Th. Morgen, H. Franke, G. Schwellach

Herstellung:
Roeske Verlag, Eschwege

Satz und Reproduktion:
Roeske Verlag, Eschwege

Druck:
E. Jungfer, 3420 Herzberg

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel),
sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Straße 7
6200 Wiesbaden
Tel.:06121-2660

Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!

Anschrift:
Roeske Verlag
HOMECOMPUTER
Fuldaer Straße 6
3440 Eschwege
Tel. Sa. Nr. 05651/8558

Anzeigenleitung:
Roeske Verlag, 3440 Eschwege

Erscheinungsweise:
Erstverkaufstag von
HOMECOMPUTER ist Anfang des Monats.

Anzeigenpreise:
Bitte Mediaunterlagen anfordern

Anzeigenannahmeschluß:
Jeweils 3 Wochen vor Erscheinungstermin

Urheberrecht:
Alle in HOMECOMPUTER veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen jeder Art (Fotokopie, Microfilm, Erfassung in Datenverarbeitungsanlagen, usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.
Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreise:
Einzelheft: 5,50 DM
Abonnement: Inland 55,-DM im Jahr
(12 Ausgaben)
Ausland: Europa 80,-DM USA 110,-DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.
Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 100,- DM pro abgedruckter Seite im Heft einverstanden sind.
Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger.
Rücksendung erfolgt nur gegen Erstattung der Unkosten. Zusendungen von Software zur Veröffentlichung sollen folgendes enthalten:
Kopierfähige Kassette oder Diskette mit dem Programm (Computer-Bezeichnung), von Drukker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreibmaschinenlistings), evtl. Bildschirmfotos von einem Probelauf und ausführliche Programmbeschreibung (Erklärung der Variablenliste, Beschreibung des Bildschirmaufbaues, Farbe, Grafik usw.). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

IVW geprüft.

Dallas

für den TI-99/4A Ext. Basic

Der Kampf um die Ewing-Millionen für 2 Personen und 1 Homecomputer

J.R. und Bobby versuchen Präsident des Ewing-Imperiums zu werden. J.R. bedient Joystick 1, Bobby Joystick Nr. 2. (Da diese meist nicht gekennzeichnet sind, entscheidet der Zufall, welcher Spieler welche Rolle übernimmt.)
Im Hauptmenü gibt es drei Möglichkeiten:
1. X-Felder gehen (nach links oder rechts)
2. X-Stimmrechte sammeln oder
3. ein Ereignis (Intrige)
Das Ereignis verschafft einem Spieler Gewinne, es schadet dem Gegenspieler oder aber beides gleichzeitig. Es gibt 4 verschiedene Arten von Feldern auf Ihrem "Dallas-Plan" (alle hellblau; J=J.R.; B =Bobby):
1. Die normalen Felder
2. Die Felder, die mit einer Firma (Texas Instruments, Barnes/Wentworth oder Harwood-Oil) verbunden sind, sowie
3. Die Felder, die mit Ewing-Oil (ganz rechts)und Southfork (ganz links) verbunden sind.
Auf den Firmen-Feldern kann man Aktien der betreffenden Firma kaufen. Sie kosten zwischen 1.000.000 $ und 3.000.000 $ (je nach Kurs). Kommt ein Spieler auf das Ewing-Oil-Feld, so werden seine Stimmrechte, sein Barvermögen sowie seine Aktien aufgelistet. Kommt ein Spieler auf das Southfork-Feld, so kann er wählen, ob ein neuer (vorübergehender) Präsident von Ewing-Oil gewählt werden soll.
In diesem Falle wird derjenige Spieler neuer Präsident, der die meisten Stimmrechte gesammelt hat. Pro Aktie kommen noch 10 Stimmrechte hinzu (z.B. zusätzlich 4 Aktien + 15 Stimmen = 55 Stimmen insgesamt). Der sogewählte Präsident erhält eine 5.000.000 $ Prämie.
Nach 52 Durchgängen (= 1 Jahr) ist es schließlich soweit: Der endgültige Präsident des Ewing-Imperiums wird bestimmt.
Präsident wird, wer am meisten Geld aus dem Startkapital von 10.000.000 $ gemacht hat. Zu dem Barvermögen kommt noch das Geld für die Aktien. 1 Aktie kann nun zwischen 2.000.000 $ und 6.000.000 $ wert sein.

Bei den Wahlmöglichkeiten ist der gelbe Pfeil mittels Joystick über die gewünschte Position zu bringen (rechts-links-Bewegung). Anschließend ist die Fire-Taste zu drücken. Dies entspricht der Maus-Idee, die Bedienung kann wirklich von jedem Computerlaien vorgenommen werden.

Bei den Warteschleifen (z.B. bei Anzeige der Bilanz) ist die Fire-Taste zu drücken, um fortzufahren. Nach einer bestimmten Zeit geht es aber auch (mit einem Ton) von selbst weiter.
Die Tastatur wird also nicht benötigt, ein bequemes Spielen ist garantiert. Das Spielprinzip ist wirklich leicht zu verstehen.
Daß das Kapital und die Stimmrechte nur angezeigt werden, wenn man sich auf dem Ewing-Oil-Feld befindet, erhöht die Spannung enorm; selbstverständlich können auch Schulden gemacht werden ("-" ist blau hervorgehoben).

Programmaufbau:
100-220: Definition
230-250: Titelbild (natürlich 3-geteilt, wie in DALLAS!)
260-280: DALLAS-Schriftzug/Copyright/Musik
290-330: Aufbau des Spielfeldes (DALLAS-Plan)
340-360: Sprite-Definition/Variablendefinierung, etc.
370-380: Tastatur (Joystick) Abfrage/Stimmrechte add.
390-410: Spielfigurbewegung (links oder rechts)
420-470: Wenn Southfork, dann Präsidentenwahl
480-510: Wenn Firmenfeld, dann Aktie kaufen
520-530: Wenn Ewing-Oil, dann Bilanz-Anzeige
540-550: Ereignis; Summe addieren bzw. substrahieren
560: Wenn Runde< 52 Figurentausch, zurück; sonst:
570-630: Endgültige Betimmung des Präsidenten
640-650: Weiter (neues Spiel) oder Ende; Musik, Stop
660-670: SUB W(R) Warte-Unterprogramm
680-720: SUB Taste (R,Z1) Joystick-Abfrage
730-910: SUB Intrige (R,Z2) Ereignis
920: SUB Texas In. (Ze,Sp) TI bzw. Texas-Zeichen
930: SUB Southfork (Ze, Sp) zeichnet die Ranch
940: SUB Bohrturm (Ze,Sp) zeichnet Harwood-Zeichen
950: SUB Förderturm (Ze,Sp) zeichnet Wendworth-Zeichen
960-970: SUB Ewing-Oil (Ze, Sp) zeichnet Ewing-Oil-Gebäude
980-1020: SUB DALLAS spielt DALLAS-Musik

Die gesamte Graphik ist hochauflösend (durch Call Char). Bildschirmfarbe: Vorspann: rot/J.R.: schwarz/Bobby: blau

Variablenliste:
Z Allgemeine Variable, z.B. für FOR...NEXT-Schleifen
Z1 Bei der Wahlmöglichkeit entweder 1, 2 oder 3
Z2 Summe bei den Ereignissen, *1E6=Betrag
Ze In den SUB-Programmen=Zeile, auch allgemein
Sp In den SUB-Programmen=Spalte, auch allgemein
Sp$(1).="J.R.".Sp$(2)="Bobby:"
R wenn R=1, dann J.R.; wenn R=2, dann Bobby
R1 =Runde
Mo().=Geld (Barvermögen)
BW() B. Wentworth-Aktien-Anzahl der jeweiligen Person
TI() Texas Instr.-Aktien-Anzahl der jeweiligen Person
HH() Harwood-Oil-Aktien-Anzahl der jeweiligen Person
S() Stimmenanzahl der jeweiligen Person/allgemein
ZU() Stimmen oder Geld der jeweiligen Person/zusammen

Die Professionalität zeigt sich u.a. in den 9 Unterprogrammen und in der Verwendung von linearen Funktionen (z.B. in den Zeilen 220-240). Diese Funktionen ersparen, in FOR..NEXT-Schleifen integriert, viele Statment's.

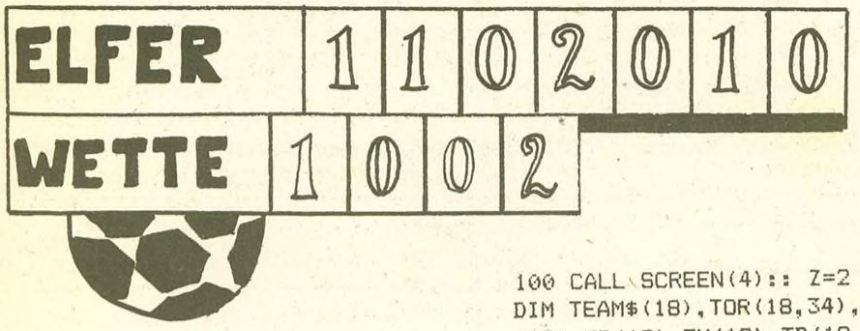
In das Programm wurden viele Gags eingebaut, angefangen mit $-Zeichen als "Wolken" im blauen Himmel von Dallas bis hin zu J.R.'s Bemerkungen über "Denver"u.ä.
Darüber hinaus finden sich noch Anspielungen auf die Beziehungen Texas Instruments zu DALLAS: Nicht nur, daß der Hauptwohnsitz von TI in Dallas liegt, auch der TI-Sprachsyntheziser hat bekanntlich einen leicht texanischen Akzent. Und deshalb läßt Ihr TI-99 J.R. und Bobby auch zu Wort kommen.
Sie werden sehen: Die Stimme wirkt enorm!
Wenn Sie jedoch keinen Sprachsyntheziser besitzen, brauchen Sie nur die CALL SAY/SPGET-Anweisungen fort zu lassen. Das Spiel läuft auch so ganz toll.

```
100 !==D A L L A S===TI99/4A(C)1984  PATRICK SCHMITZ, AMZEHNTHOF4,5480 REM-UNKEL
BACH*ANFRAGEN N U R SCHRIFTL.!*
120 CALL CLEAR :: CALL SCREEN(2):: CALL CHAR(64,"3C4299A1A199423C"):: CALL CHAR(
94,"000000FFFF0000000000000000003030")
130 CALL CHAR(34,"383838F8F8F8000000000000FFFFFF0000"):: CALL CHAR(91,"0038545038
4543800000000030102000000000007C000000")
140 CALL CHAR(32,"0001000000000080")
3C18FFFFFF7F562000001818181818181818")
150 CALL CHAR(37,"383838383838383818 3C7EFFFF7E
R(1,1,31,160):: CALL HCHAR(6,1,39,32):: CALL SAY("U R IN THE U S A")
160 DATA 000003071F1F09090B0D0F090B0D1F1117191F1117193F2121FFFF2127393F21,233D7F
414779FF414147F9FF8181818100000C0F0F02020A060E020A060F010
170 DATA D030F010D030F80808FEFE08C83BF808887BFC04C43CFC0404C43EFE02020202,000000
0000000607070604040408080E8E8FCFFFFFFFFFFFF00000000073FFFF8
180 DATA DC1C2A2A2A2A2A4949494988888FFFF103070F8F8F8787C3C3C1810008080808,080808
888888C8C8181818181818181818FFFEFCF8F0E0C0800103070F1F3F7FFF
190 DATA 1F3F7FFF80FFE6E63FFFFFFF00FF77779FCFE7F700F7F7F7EFC7C7FF00FFE7E7,309794
D4F7F0F7F700FF6464FF00DBDB3DDB585FD01EDEDEE0E0F0F018786060
200 DATA 0000000020707 0F0000000000000F1F000001030707FEFF0000F3F93780EF6F,10387C
FCFE016767000000000000BFDF0000000000000B0C000FF18FF00000000
210 DATA 0303030303030303E0E0E0E0FBFFFFFF0000000020E0E0E0FF7F3F3F3F130000,FFFFFF
FFFFFFFEFCF0F8F0F0C00000003B1C0000000000001818181BFF7E3C18
220 FOR Z=96 TO 143 STEP 4 :: READ Z$ :: CALL CHAR(Z,Z$):: NEXT Z :: FOR Z=1 TO
6 STEP 5 :: CALL VCHAR(Z,11,15.4*Z+24.6,2.6*Z+3.4)
230 CALL VCHAR(Z,23,15.4*Z+24.6,2.6*Z+3.4)
):: CALL BOHRTURM(.2*Z+11.8,.8*Z+24.2)
240 CALL BOHRTURM(.2*Z+17.8,.8*Z+25.2):: CALL EWINGOIL(.2*Z+16.8,.6*Z+4.4):: NEX
T Z :: CALL SOUTHFORK(11,14):: CALL SPGET("#TEXAS INSTRUMENTS",Z$):: CALL SCREEN
(9)
250 CALL COLOR(0,8,8,1,8,3,2,2,8,12,16,3,13,16,3,14,2,3):: FOR Z=9 TO 11 :: CALL
 COLOR(Z,2,3):: NEXT Z :: FOR Z=3 TO 8 :: CALL COLOR(Z,12,3):: NEXT Z
260 CALL SAY("U R IN",SEG$(Z$,1,94)):: Z$="DALLAS" :: FOR Z=1 TO 6 :: CALL SPRIT
E(#Z,ASC(SEG$(Z$,Z,1)),12,42+Z*3.3,88+Z*12):: NEXT Z :: DISPLAY AT(14,10)SIZE(10
):"DAS SPIEL"
270 DISPLAY AT(15,11)SIZE(10):"DER EWINGS" :: DISPLAY AT(20,13)SIZE(5):"@1984" :
: DISPLAY AT(21,12)SIZE(7):"PATRICK" :: DISPLAY AT(22,12)SIZE(7):"SCHMITZ" :: CA
LL MAGNIFY(2)
280 CALL SAY("YES, U R  J R"):: CALL DALLAS :: CALL CLEAR :: FOR Z=3 TO 8 :: CAL
L COLOR(Z,2,3):: NEXT Z :: CALL DELSPRITE(ALL):: CALL MAGNIFY(1):: DISPLAY AT(1,
1):"DALLAS  DAS SPIEL DER EWINGS"
290 CALL HCHAR(18,1,94,32):: CALL SOUTHFORK(2,25):: CALL EWINGOIL(7,3):: CALL HC
HAR(8,6,38,21):: CALL HCHAR(9,6,38,21):: FOR Z=9 TO 21 STEP 6
300 CALL VCHAR(5,Z,37,3):: CALL VCHAR(10,Z,37,3):: NEXT Z :: CALL HCHAR(9,28,34)
:: CALL HCHAR(8,27,34):: CALL FOERDERTURM(3,8):: CALL BOHRTURM(12,15)
310 CALL TEXASIN(3,20):: CALL VCHAR(5,27,37,3):: CALL VCHAR(5,28,37,4):: CALL HC
HAR(9,27,35):: CALL VCHAR(8,5,35,2)
320 DISPLAY AT(3,1)SIZE(3):"EN_" :: DISPLAY AT(4,1)SIZE(3):"OIL" :: DISPLAY AT(1
3,5)SIZE(6):"BARNES" :: DISPLAY AT(14,5)SIZE(6):"WENTW_" :: DISPLAY AT(13,17)SIZ
E(5):"TEXAS"
330 DISPLAY AT(14,17)SIZE(5):"INST_" :: DISPLAY AT(3,11)SIZE(5):"HARM_" :: DISPL
AY AT(4,12)SIZE(3):"OIL" :: DISPLAY AT(5,27)SIZE(2):"SF" :: CALL SAY("#READY TO
START")
340 CALL SPRITE(#1,74,2,56,200,#2,66,2,64,200,#3,143,12,136,64):: CALL SAY("YOUR
 CHOICE"):: RANDOMIZE
350 R=1 :: APOS(1),APOS(2)=200 :: SP$(1)="J_R_ :" :: SP$(2)="BOBBY:" :: DISPLAY
AT(17,1)SIZE(6):SP$(1):: MO(1),MO(2)=10E6
360 ZE=INT(RND*8)+2 :: SP=INT(RND*8)+2 :: Z=INT(2000001*RND)+1E6 :: DISPLAY AT(1
9,1):ZE;"FELDER OD_";SP;"STIMMEN OD_:": : : : : : :: CALL VCHAR(20,2,117,5)
370 CALL VCHAR(20,31,117,5):: CALL INTRIGE(R,Z2):: CALL TASTE(R,Z1):: IF Z1=.2 T
HEN 540 ELSE IF Z1=0 THEN 390 ELSE CALL HCHAR(19,1,32,192):: S(R)=S(R)+SP :: GOT
O 560
390 CALL HCHAR(19,1,32,192):: DISPLAY AT(19,1):"NACH LINKS  OD_ RECHTS" :: DISPL
AY AT(21,4):"UM";ZE;"FELDER" :: CALL TASTE(R,Z1):: IF Z1=.2 THEN 390
400 IF Z1=0 AND APOS(R)-ZE*8>=40 THEN APOS(R)=APOS(R)-ZE*8 :: CALL LOCATE(#R,R*8
+48,APOS(R)):: GOTO 420 ELSE IF Z1=0 THEN APOS(R)=40 :: CALL LOCATE(#R,R*8+48,40
):: GOTO 420
410 IF Z1=.1 AND APOS(R)+ZE*8<=200 THEN APOS(R)=APOS(R)+ZE*8 :: CALL LOCATE(#R,R
*8+48,APOS(R))ELSE APOS(R)=200 :: CALL LOCATE(#R,R*8+48,200)
420 IF APOS(R)<>200 THEN 480 ELSE CALL HCHAR(19,1,32,192):: DISPLAY AT(19,6):"JA
":TAB(18);"NEIN" :: DISPLAY AT(21,1):"SOLL DER PRAESIDENT VON     EWINGJOIL NEUE
 GEWAEHLT"
430 DISPLAY AT(23,1):"WERDEN ???" :: CALL TASTE(R,Z1):: IF Z1=.2 THEN CALL SAY("
#TRY ABAIN"):: GOTO 430 ELSE IF Z1=.1 THEN CALL SAY("O K WHY NOT"):: GOTO 560
440 FOR Z=1 TO 2 :: DISPLAY AT(19,1):"STIMMRECHTE V_ ";SP$(Z);"STIMMEN":BW(Z);"B
_WENTWORTHJAKTIEN:";BW(Z)*10:HH(Z);"HARWOODOILJAKTIEN :";HH(Z)*10:TI(Z);"TEXAS I
NST_J"
450 DISPLAY AT(22,16):"AKTIEN:";TI(Z)*10:" SONSTIGE STIMMRECHTE:";S(Z):: ZU(Z)=B
W(Z)*10+HH(Z)*10+TI(Z)*10+S(Z):: DISPLAY AT(24,Z*15-14):"ZUS_";SP$(Z);ZU(Z)
460 CALL W(R):: NEXT Z :: IF ZU(1)>ZU(2)THEN CALL SAY("J R IS FIRST"):: SP=1 ELS
E IF ZU(2)>ZU(1)THEN SP=2 ELSE DISPLAY AT(19,1): :"J_R_UND BOBBY HABEN GLEICH  V
IELE STIMMEN_": : : :: CALL W(R):: GOTO 560
470 DISPLAY AT(19,1): :SEG$(SP$(SP),1,5)&" IST NEUER PRAESIDENT  VON EWINGJOIL_
ER ERHAELT  5_000_000I_": : : :: CALL W(R):: MO(SP)=MO(SP)+5E6 :: GOTO 560
480 IF APOS(R)=64 THEN DISPLAY AT(22,1):"BARNES WENTWORTH" :: CALL FOERDERTURM(2
0,27):: GOSUB 510 :: IF Z1=.1 THEN CALL SAY("U MIGHT MAKE AN ERROR"):: GOTO 560
ELSE BW(R)=BW(R)+1 :: MO(R)=MO(R)-Z :: GOTO 560
490 IF APOS(R)=112 THEN DISPLAY AT(22,1):"HOLLY HARWOODJOIL" :: CALL BOHRTURM(19
,29):: GOSUB 510 :: IF Z1=.1 THEN CALL SAY("OH NO"):: GOTO 560 ELSE HH(R)=HH(R)+
1 :: DMO(R)=MO(R)-Z :: GOTO 560
500 IF APOS(R)=160 THEN DISPLAY AT(22,1):"TEXAS INSTRUMENTS INC_" :: GOSUB 510 :
: IF Z1=.1 THEN 560 ELSE TI(R)=TI(R)+1 :: MO(R)=MO(R)-Z :: CALL SAY("#TEXAS INST
RUMENTS",,"IS TOP"):: GOTO 560 ELSE 520
510 DISPLAY AT(19,1)SIZE(26):"   KAUFEN OD_ NICHT KAUFEN" :: DISPLAY AT(21,1)SIZ
E(20):"AKTIEN VON:" :: DISPLAY AT(24,1)SIZE(16):"PREIS:";Z;"[" :: CALL TASTE(R,Z
1):: IF Z1=.2 THEN 510 ELSE RETURN
520 IF APOS(R)<>40 THEN 560 ELSE DISPLAY AT(19,1):"STIMMRECHTE VON ";SP$(R);BW(R
)*10+HH(R)*10+TI(R)*10+S(R):"KAPITAL VON ";SP$(R);MO(R);"["
530 DISPLAY AT(21,1): :BW(R);"B_WENTWORTHJAKTIEN":HH(R);"HARWOODOILJAKTIEN":TI(R
):"TEXAS INST_JAKTIEN" :: CALL W(R):: CALL SAY("#GOOD WORK"):: GOTO 560
```

```
540 IF R=1 AND Z2=0 THEN MO(2)=MO(2)-1E6 ELSE IF R=1 AND Z2<0 THEN MO(1)=MO(1)-Z
2*1E6 :: MO(2)=MO(2)+Z2*1E6 ELSE IF R=1 THEN MO(1)=MO(1)+Z2*1E6
550 IF R=2 AND Z2=0 THEN MO(1)=MO(1)-1E6 ELSE IF Z2<0 AND R=2 THEN MO(2)=MO(2)-Z
2*1E6 :: MO(1)=MO(1)+Z2*1E6 ELSE IF R=2 THEN MO(2)=MO(2)+Z2*1E6
560 R1=R1+1 :: IF R=2 THEN IF R1<52 THEN DISPLAY AT(17,1)SIZE(6):SP$(1):: R=1 ::
CALL SCREEN(2):: GOTO 360 ELSE 570 ELSE DISPLAY AT(17,1)SIZE(6):SP$(2):: R=2 ::
CALL SCREEN(5):: GOTO 360
570 CALL HCHAR(17,3,32,6):: CALL HCHAR(19,1,32,192):: CALL LOCATE(#1,56,200,#2,6
4,200):: DISPLAY AT(19,2):"ES IST SOWEIT___"
580 DISPLAY AT(20,1):" WER BIS JETZT AM MEISTEN   GELD AUS DEM STARTKAPITAL   G
EMACHT HAT\WIRD ENTGUELTIJ GER PRAESIDENT V_ EWINGOIL_" :: CALL SAY("IT IS TIME
TO SUM UP")
590 CALL W(R):: CALL SCREEN(9):: FOR Z1=1 TO 2 :: FOR Z=1 TO 3 :: S(Z)=INT(RND*4
000001)+2E6 :: NEXT Z :: DISPLAY AT(20,1):BW(Z1);"B_W_AKTIEN JE:"S(1);"[":HH(Z1);"HAR_AKTIEN JE:";S(
600 DISPLAY AT(20,1):BW(Z1);"B_W_AKTIEN JE:";S(1);"[";HH(Z1);"HAR_AKTIEN JE:";S(
2);"[":TI(Z1);"TI JAKTIEN JE:";S(3);"[".
610 DISPLAY AT(23,1):"SONSTIG_ KAPITAL:";MO(Z1);"[" :: CALL W(R):: ZU(Z1)=BW(Z1)
*S(1)+HH(Z1)*S(2)+TI(Z1)*S(3)+MO(Z1):: NEXT Z1
620 IF ZU(1)>ZU(2)THEN SP=1 ELSE IF ZU(1)<ZU(2)THEN SP=2 ELSE DISPLAY AT(19,1):
:" J_R_UND BOBBY WAREN GLEICH  BUT_SIE MUESSEN EWINGOIL  GEMEINSAM LEITEN_": :
:: GOTO 640
630 DISPLAY AT(19,1): :"  ";SEG$(SP$(SP),1,5);" IST PRAESIDENT DES:"      EWING
IMPERIUMS_": : : :: CALL EWINGOIL(20,2):: CALL EWINGOIL(20,30)
640 FOR Z=1 TO 2 :: DISPLAY AT(22+Z,2)SIZE(25):"  ";SP$(Z);ZU(Z);"[" :: NEXT Z :
: CALL DALLAS :: CALL SAY("YOU DID IT VERY WELL","#GOOD WORK"):: CALL HCHAR(19,
1,32,192)
650 DISPLAY AT(19,2):"NEUES SPIEL  KEIN NEU_SPIEL" :: CALL TASTE(R,Z):: IF Z=0 T
HEN RUN 340 ELSE IF Z=.1 THEN CALL DALLAS :: CALL SAY("O K, BUT I M SURE J R WIL
L RETURN, GOODBYE")ELSE 650
655 CALL CLEAR :: CALL SCREEN(2):: PRINT "DAS ORIGINAL":"(ALSO *D A L L A S* )":
"IST JA WOHL 1000 MAL BESSER":"ALS DIE KOPIE('DENVER'+CO.)."": :"J.R.: HE! HE! HE
!": : : :
660 SUB W(R):: CALL COLOR(#3,1):: FOR Z=1 TO 200 :: CALL KEY(R,ZE,SP):: IF SP<>-
1 AND ZE=18 THEN CALL SOUND(-100,1568,4,784,8,392,8):: SUBEXIT
670 NEXT Z :: CALL SOUND(-100,880,4,220,8,440,8):: CALL SAY("NOW WE SHOULD GO ON
"):: SUBEND
680 SUB TASTE(R,Z1):: CALL COLOR(#3,12)
690 CALL JOYST(R,ZE,SP):: IF ZE=4 AND Z1=0 THEN CALL LOCATE(#3,136,168):: Z1=.1
ELSE IF ZE=4 AND Z1=.1 THEN CALL LOCATE(#3,136,240):: Z1=.2
700 IF ZE=-4 AND Z1=.2 THEN CALL LOCATE(#3,136,168):: Z1=.1 ELSE IF ZE=-4 THEN C
ALL LOCATE(#3,136,64):: Z1=0
710 FOR Z=1 TO 50 :: NEXT Z :: CALL KEY(R,ZE,SP):: IF ZE<>18 THEN 690 ELSE CALL
SOUND(-100,1047,8,523,4,262,8):: IF RND>.92 THEN CALL SAY("#NICE TRY")ELSE IF RN
D<.08 THEN CALL SAY("O K")
720 SUBEND
730 SUB INTRIGE(R,Z2):: Z2=0 :: IF R=2 THEN 820 ELSE ON INT(RND*7)+1 GOTO 740,75
0,770,780,790,800,810
740 Z$="J_R_ STELLT FEST\ DASS GEI  WISSE\EINKUENFTE VON BOBBY  DEM FINANZAMT UN
BEKANNT  BLIEBEN_ BOBBY TUT GUT DARANJ_R_ 1_000_000[ ZU ZAHLEN___" :: Z2=-1 :
: GOTO 910
750 Z$="BOBBY HAT DIE GLEICHE BLUTJ GRUPPE WIE DAS KIND VON  KARIN\ SEINER EX
FREUNDIN___ J_R_ WIRD MIT DIESEM WISSEN UM 1_000_000[ REICHER___" :: Z2=-1 :: GO
TO 910
770 Z$="J_R_ WEISS\ DASS BOBBY EIN  MITGLIED DER TEXASJENERGIEJ KOMMISSION BESTO
CHEN HAT_  1_000_000 LEICHT VERDIENTE  DOLLAR FUER J_R_ ___" :: Z2=-1 :: GOTO 9
10
780 Z$="J_R_KOMMT ZU BOBBYS  GEHEIMJPAPIEREN_SEIN NAECHSTER AUFJTRITT KOENNTE NI
CHT IM FERNJSEHEN\SONDERN VOR GERICHT  SEIN_J_R_BEKOMMT 1_000_000[_" :: Z2=-1 :
: GOTO 910
790 Z$="J_R_S PROBEBOHRUNGEN SIND  FUENDIG GEWORDEN_ DAS OEL  SPRUDELT NUR SO_
J_R_ KASJ  SIERT 3_000_000[_" :: Z2=3 :: GOTO 910
800 Z$="SPEKULATIVE LANDBESCHAEFTE  MACHEN SICH BEZAHLT_ J_R_  BEKOMMT 2_000_00
0[_" :: Z2=2 :: GOTO 910
810 Z$="BOBBY SCHENKT PAM EIN AUTO  UND SPENDIERT LUCY EINEN URJLAUB AM STRAND\
WO SIE SICH IN D_ SONNE SCHMOREN LAESST_ZUSAMMEN 1_000_000[___" :: GOTO 910
820 ON INT(RND*7)+1 GOTO 830,840,850,860,870,880,890
830 Z$="BOBBY ERFAEHRT VON J_R_S  GEHEIMEN BOHRVORHABEN IN  FERNOST_ J_R_ IS
T DIESES  GEHEIMNIS 1_000_000[ WERT___" :: Z2=-1 :: GOTO 910
840 Z$="SUE ELLENS ALTE LIEBESBRIEJ FE WAREN HEISS GENUG\ UM  BRANDBLASEN ZU E
RZEUGEN_J_R_ZAHLT 1_000_000[\DAMIT BOBBYSIE NICHT VERDOEFFENTLICHT_".:: Z2=-1 ::
GOTO 910
850 Z$="J_R_ ZAHLT BOBBY 1_000_000[\WEIL ER VON SEINER BEZIEHUNGZU MARILEE STONE
ERFAHREN  HAT_" :: Z2=-1 :: GOTO 910
860 Z$="SUE ELLEN WUERDE NACHTS VOR SCHAM ROT LEUCHTEN\WENN IHREALTEN LIEBESBRIF
E JE BEKANNTWUERDEN_J_R_ZAHLT 1_000_000[UM DAS ZU VERHINJDERN_" :: GOTO 910
870 Z$="CLIFF BARNES LEITET UNTERSUJCHUNGEN WEGEN UNERLAUBTER  PREISABSPRACHEN
EIN_ J_R_  ZAHLT IHM 1_000_000[___" :: GOTO 910
880 Z$="J_R_ SCHENKT MISS ELLIE  GRUNDSTUECKE ZUM GEBURTSTAG\DIE AN SOUTHFORK
GRENZEN_SIEIST GLUECKLICH UND ER UM  1_000_000[ AERMER" :: GOTO 910
890 Z$="DIVIDENDEN WERDEN FAELLIG_  BOBBY KANN 2_000_000[  KASSIEREN_" :: Z
2=2 :: GOTO 910
910 DISPLAY AT(20,1):Z$ :: SUBEND
920 SUB TEXASIN(ZE,SP):: FOR Z=0 TO 2 :: CALL HCHAR(ZE,SP+Z,136+Z):: CALL HCHAR(
ZE+1,SP+Z,139+Z):: NEXT Z :: CALL HCHAR(ZE+2,SP+1,142):: SUBEND
930 SUB SOUTHFORK(ZE,SP):: FOR Z=SP TO SP+7 :: CALL HCHAR(ZE+1,Z,120+Z-SP):: NEX
T Z :: FOR Z=SP+1 TO SP+7 :: CALL HCHAR(ZE,Z,127+Z-SP):: NEXT Z :: CALL HCHAR(ZE
+2,SP,135,8):: SUBEND
940 SUB BOHRTURM(ZE,SP):: FOR Z=ZE TO ZE+5 :: CALL HCHAR(Z,SP,96+Z-ZE):: CALL HC
HAR(Z,SP+1,102+Z-ZE):: NEXT Z :: SUBEND
950 SUB FOERDERTURM(ZE,SP):: FOR Z=ZE TO ZE+2 :: CALL HCHAR(Z,SP,108+Z-ZE):: CAL
L HCHAR(Z,SP+1,111+Z-ZE):: CALL HCHAR(Z,SP+2,114+Z-ZE):: NEXT Z :: SUBEND
960 SUB EWINGOIL(ZE,SP):: FOR Z=1 TO 2 :: CALL VCHAR(ZE,SP-1+Z,40+Z,6-Z):: CALL
HCHAR(ZE+(-5*Z+9),SP-Z+2,117+Z):: NEXT Z :: CALL HCHAR(ZE-1,SP+1,43):: SUBEND
980 SUB DALLAS
990 DATA 200,523,300,659,200,523,300,784,200,523,300,659,200,587,200,659,300,523
,300,392,300,523,300,880,300
1000 DATA 784,200,659,200,698,400,784,100,523,200,523,300,880,300,784,200,659,20
0,698,300
1010 DATA 784,200,587,200,659,300,523,300,392,300,523,200,659,200,698,300,587,20
0,784,300,784
1020 RESTORE 990 :: FOR Z=1 TO 33 :: READ Z1,ZE :: CALL SOUND(Z1,ZE,2,ZE/2,20,ZE
*2,25):: NEXT Z :: SUBEND ! DER BESTEN TV-SERIE DER WELT + ALLER ZEITEN -DALLAS
- SOWIE J.R.-L.HAGMAN- GEWIDMET.
```

BUNDESLIGA 1984/85

für den TI 99/Extended Basic



Ein Programm für alle Fußballfreunde und besonders für Tototipper.
Wenn Sie das Programm starten, erscheint ein Menü. Haben Sie den Spieltag gewählt, fragt Sie das Programm, welchen Spieltag Sie haben möchten. Ist der Spieltag eingegeben, erscheinen die Spielpaarungen. Sie können Tore eingeben, indem Sie die beiden Anfangsbuchstaben der Mannschaften tippen. Der Tototipper kann sich auch vom Computer beraten lassen: er braucht nur den Spieltag einzugeben. Das Programm ist auch für die nächsten Jahre zu gebrauchen, wenn man die Zeilen 110-120 so verändert, daß die Auf- und Absteiger berücksichtigt werden und in den Zeilen 400-560 in Spielpaarungen geändert werden.

Variablen-Liste

Schleifen-Variablen
I,X,S1,S2,Q
Z= Zeile
SPA= Spalte
TEAM= Mannschaft
TOR= TORE (Mannschaft, Spieltag)
PUN= Punkte
TP= Plustore
TM= Minustore
TD= Tor-Differenz
KL= Klassenwert der Mannschaft
SP= Spieltag
RUECK= Rückrunde
TA= Tabelle
TIP= Computer Tip
CT= Computer Tip (Tore)
CTT= Computer Tip (Tore)
ZEIT= Zeit (für die Tabellenberechnung)
110-120: Data-Zeilen: Mannschaften
400-560: Data-Zeilen: Spieltagpaarungen
770-870: Tabellenberechnung
880-970: Tabellensortierung
1210-1260: Computertipberechnung

```
100 CALL SCREEN(4):: Z=2 ::
DIM TEAM$(18),TOR(18,34),PUN
(18),TP(18),TM(18),TD(18
),KL(18)
110 DATA WERD.BREMEN,UERDING
EN,A.BIELEFELD,B.MUENCHEN,E.
B'SCHWEIG,1.FC KOELN,K'L
AUTERN,STUTTGART,KARLSRUHE,W
ALDHOF M.
120 DATA LEVERKUSEN,DUESSELD
ORF,VFL BOCHUM,FRANKFURT,BR.
GLADBACH,SCHALKE 04,BO.D
ORTMUND,HAMBURG
130 DATA 5,13,10,2,15,8,12,3
,18,12,6,11,17,9,4,16,14,1
140 RESTORE 130 :: FOR I=1 T
O 18 :: READ KL(I):: NEXT I
150 RESTORE 110 :: FOR I=1 T
O 18 :: READ TEAM$(I):: NEXT
 I
160 DISPLAY AT(4,2)ERASE ALL
:"***BUNDESLIGA***1984/85***
" :: DISPLAY AT(12,3):"1
  TABELLE" :: DISPLAY AT(14,
3)BEEP:"2  SPIELTAG" :: DISP
LAY AT(16,3):"3  COMPUTE
R TIP"
170 DISPLAY AT(10,1):"TASTE"
 :: DISPLAY AT(18,3):"4  ERG
EBNISSE EINLESEN" :: DIS
PLAY AT(20,3):"5  ERGEBNISSE
 SPEICHERN" :: DISPLAY AT(22
,3):"6  ENDE"
180 CALL KEY(0,K,S):: IF S=0
 THEN 180 ELSE IF K=49 THEN
740 ELSE IF K=50 THEN 19
0 ELSE IF K=51 THEN TIP=1 ::
 GOTO 190 ELSE IF K=52 THEN
1060 ELSE IF K=53 THEN 1
120 ELSE IF K=54 THEN 1270 E
LSE GOTO 180
190 DISPLAY AT(12,2)ERASE AL
L BEEP:"GEBEN SIE DEN SPIELT
AG EIN" :: DISPLAY AT(14
,11):"(1-34)"
200 ACCEPT AT(16,13)VALIDATE
(DIGIT)SIZE(2)BEEP:SP :: IF
SP<1 OR SP>34 THEN 200
210 IF SP>17 THEN SP=SP-17 :
: RUECK=1 ELSE RUECK=0
220 ON SP GOTO 230,240,250,2
60,270,280,290,300,310,320,3
30,340,350,360,370,380,3
90
230 RESTORE 400 :: GOTO 570
240 RESTORE 410 :: GOTO 570
250 RESTORE 420 :: GOTO 570
260 RESTORE 430 :: GOTO 570
270 RESTORE 440 :: GOTO 570
280 RESTORE 450 :: GOTO 570
290 RESTORE 460 :: GOTO 570
300 RESTORE 470 :: GOTO 570
310 RESTORE 480 :: GOTO 570
320 RESTORE 490 :: GOTO 570
330 RESTORE 500 :: GOTO 570
340 RESTORE 510 :: GOTO 570
350 RESTORE 520 :: GOTO 570
360 RESTORE 530 :: GOTO 570
370 RESTORE 540 :: GOTO 570
380 RESTORE 550 :: GOTO 570
390 RESTORE 560 :: GOTO 570
400 DATA 1,2,3,4,5,6,7,8,9,1
0,11,12,13,14,15,16,17,18
410 DATA 18,15,14,11,12,9,10
,7,8,5,6,3,4,1,2,17,16,13
420 DATA 1,6,3,8,5,10,7,12,9
,14,11,16,2,4,17,15,13,18
430 DATA 18,11,14,7,12,5,10,
3,8,1,6,2,4,17,15,13,16,9
440 DATA 1,10,3,12,5,14,7,16
,9,18,11,15,4,6,2,8,17,13
450 DATA 18,7,14,3,12,1,10,2
,8,4,6,17,13,11,15,9,16,5
460 DATA 1,14,3,16,5,18,7,15
,9,13,6,8,4,10,2,12,17,11
470 DATA 18,3,14,2,12,4,10,6
,8,17,11,9,13,7,15,5,16,1
480 DATA 1,18,3,15,5,13,7,11
,8,10,6,12,4,14,2,16,17,9
490 DATA 18,2,14,6,12,8,10,1
7,9,7,11,5,13,3,15,1,16,4
500 DATA 1,13,3,11,5,9,10,12
,8,14,6,16,4,18,2,15,17,7
510 DATA 18,6,14,10,12,17,7,
5,9,3,11,1,13,2,15,4,16,8
520 DATA 1,9,3,7,12,14,10,16
,8,18,6,15,4,13,2,11,17,5
530 DATA 18,10,14,17,5,3,7,1
,9,2,11,4,13,6,15,8,16,12
540 DATA 1,5,3,17,12,18,10,1
5,8,13,6,11,4,9,16,14,2,7
550 DATA 18,14,3,1,5,2,9,6,1
1,8,15,12,17,16,7,4,13,10
560 DATA 1,17,14,15,12,13,10
,11,8,9,6,7,4,5,2,3,16,18
570 IF TA THEN RETURN ELSE I
F RUECK THEN SP=SP+17
580 IF TIP THEN A$="COMPUTER
 TIP" ELSE A$=STR$(SP)&".SPI
ELTAG"
590 DISPLAY AT(1,9)ERASE ALL
 BEEP:A$ :: IF RUECK THEN 65
0
600 FOR I=1 TO 17 STEP 2 ::
READ C,CC :: GOSUB 1200 :: Z
=Z+2 :: DISPLAY AT(Z,1):
TEAM$(C):: DISPLAY AT(Z,13):
"-";TEAM$(CC)
610 IF TIP THEN A$=STR$(CT)&
":"&STR$(CTT)ELSE A$=STR$(TO
R(C,SP))&":"&STR$(TOR(CC
,SP))
620 DISPLAY AT(Z,26):A$ :: N
EXT I :: Z=2
630 DISPLAY AT(24,6):"WEITER
 MIT >ENTER<" :: IF TIP THEN
 1190 ELSE DISPLAY AT(22
```

```
,6):"TOR EINGABE  >T<"
640 CALL KEY(0,K,S):: IF K=8
4 THEN 680 ELSE IF K<>13 THE
N 640 ELSE 160
650 FOR I=1 TO 17 STEP 2 ::
READ C,CC :: GOSUB 1200 :: Z
=Z+2 :: DISPLAY AT(Z,1):
TEAM$(CC):: DISPLAY AT(Z,13)
:"-";TEAM$(C)
660 IF TIP THEN A$=STR$(CTT)
&":"&STR$(CT)ELSE A$=STR$(TO
R(CC,SP))&":"&STR$(TOR(C
,SP))
670 DISPLAY AT(Z,26):A$ :: N
EXT I :: Z=2 :: GOTO 630
680 DISPLAY AT(12,3)ERASE AL
L:"WELCHER CLUB HAT EIN TOR"
 :: DISPLAY AT(14,9):"GE
SCHOSSEN?" :: DISPLAY AT(16,
5):"(2 ANFANGS ZEICHEN)"
690 ACCEPT AT(18,14)VALIDATE
(UALPHA,"1.'")SIZE(2)BEEP:A$
700 FOR I=1 TO 17 STEP 2 ::
IF A$=SEG$(TEAM$(I),1,2)THEN
 TOR(I,SP)=TOR(I,SP)+1 :
: GOTO 210
710 IF A$=SEG$(TEAM$(I+1),1,
2)THEN TOR(I+1,SP)=TOR(I+1,S
P)+1 :: GOTO 210
720 NEXT I
730 DISPLAY AT(24,1)BEEP:"FA
LSCHE EINGABE" :: FOR I=1 TO
 300 :: NEXT I :: GOTO 6
80
740 FOR I=1 TO 18 :: IF TOR(
I,SP)<>0 THEN 770
750 NEXT I
760 DISPLAY AT(10,2)ERASE AL
L BEEP:"KEINE ERGEBNISSE VOR
HANDEN!" :: FOR I=1 TO 3
00 :: NEXT I :: GOTO 160
770 DISPLAY AT(12,4)ERASE AL
L:"BITTE UM ETWAS GEDULD!" :
: TA=1 :: ZEIT=INT(SP*3.
2+10):: DISPLAY AT(14,7):"CA
.";ZEIT;"SEKUNDEN"
780 FOR X=1 TO SP :: IF X>17
 THEN RUECK=1 ELSE RUECK=0
790 IF RUECK THEN J=X-17 ELS
E J=X
800 ON J GOSUB 230,240,250,2
60,270,280,290,300,310,320,3
30,340,350,360,370,380,3
90
810 FOR I=1 TO 17 STEP 2
820 IF RUECK THEN READ CC,C
ELSE READ C,CC
830 IF TOR(C,X)>TOR(CC,X)THE
N PUN(C)=PUN(C)+2 :: GOTO 86
0
840 IF TOR(C,X)=TOR(CC,X)THE
N PUN(C)=PUN(C)+1 :: PUN(CC)
=PUN(CC)+1 :: GOTO 860
850 PUN(CC)=PUN(CC)+2 :: GOT
O 860
860 TM(C)=TM(C)+TOR(CC,X)::
TM(CC)=TM(CC)+TOR(C,X):: TP(
C)=TP(C)+TOR(C,X):: TP(C
C)=TP(CC)+TOR(CC,X):: TD(C)=
TP(C)-TM(C):: TD(CC)=TP(CC)-
TM(CC)
870 NEXT I :: GOSUB 1180 ::
NEXT X
880 FOR S1=1 TO 17
890 FOR S2=S1+1 TO 18
900 IF PUN(S1)>PUN(S2)THEN 9
60 ELSE IF PUN(S1)=PUN(S2)TH
EN 930
910 G=PUN(S1):: G$=TEAM$(S1)
:: F=TP(S1):: PUN(S1)=PUN(S2
):: TEAM$(S1)=TEAM$(S2):
: TP(S1)=TP(S2):: PUN(S2)=G
:: TEAM$(S2)=G$ :: TP(S2)=F
920 K=TD(S1):: TD(S1)=TD(S2)
:: TD(S2)=K :: WE=TM(S1):: T
M(S1)=TM(S2):: TM(S2)=WE
 :: GOTO 960
930 IF TD(S1)<TD(S2)THEN 940
 ELSE 960
940 M=TD(S1):: O=PUN(S1):: J
$=TEAM$(S1):: L=TP(S1):: TD(
S1)=TD(S2):: PUN(S1)=PUN
(S2):: TEAM$(S1)=TEAM$(S2)::
 TP(S1)=TP(S2):: TD(S2)=M ::
 PUN(S2)=O :: TEAM$(S2)=
J$ :: TP(S2)=L
950 WE=TM(S1):: TM(S1)=TM(S2
):: TM(S2)=WE
960 NEXT S2
970 NEXT S1
980 Z=5 :: TA=0 :: A$="TABEL
LE ("&STR$(SP)&".SPIELTAG)"
:: DISPLAY AT(1,5)BEEP E
RASE ALL:A$ :: DISPLAY AT(3,
18):"TORE   PUN"
990 FOR I=1 TO 18 :: IF I>9
THEN S=1 ELSE S=2
1000 DISPLAY AT(Z,S):I;TEAM$
(I):: DISPLAY AT(Z,17):USING
 "##":TP(I):: DISPLAY AT
(Z,19):"-" :: DISPLAY AT(Z,2
0):USING "##":TM(I)
1010 DISPLAY AT(Z,23):USING
"###":TD(I):: IF TD(I)<=0 TH
EN GOTO 1030 ELSE IF TD(
I)>0 AND TD(I)<10 THEN SPA=2
4 ELSE SPA=23
1020 DISPLAY AT(Z,SPA)SIZE(1
):"+"
1030 DISPLAY AT(Z,27):USING
"##":PUN(I):: Z=Z+1 :: NEXT
I :: DISPLAY AT(24,6):"W
EITER MIT >ENTER<" :: Z=2
1040 CALL KEY(0,K,S):: IF K<
>13 THEN 1040 ELSE 1050
1050 FOR I=1 TO 18 :: PUN(I)
=0 :: TM(I)=0 :: TP(I)=0 ::
TD(I)=0 :: NEXT I :: RES
TORE 110 :: GOTO 150
1060 CALL CLEAR :: OPEN #1:"
CS1",INTERNAL,INPUT ,FIXED 1
92 :: INPUT #1:SP
1070 FOR I=1 TO SP
1080 FOR Q=1 TO 18 STEP 9
1090 INPUT #1:TOR(Q,I),TOR(Q
+1,I),TOR(Q+2,I),TOR(Q+3,I),
TOR(Q+4,I),TOR(Q+5,I),TO
R(Q+6,I),TOR(Q+7,I),TOR(Q+8,
I)
1100 NEXT Q
1110 NEXT I :: CLOSE #1 :: G
OTO 160
1120 CALL CLEAR :: OPEN #1:"
CS1",INTERNAL,OUTPUT,FIXED 1
92 :: PRINT #1:SP
1130 FOR I=1 TO SP
1140 FOR Q=1 TO 18 STEP 9
1150 PRINT #1:TOR(Q,I);TOR(Q
+1,I),TOR(Q+2,I),TOR(Q+3,I),
TOR(Q+4,I),TOR(Q+5,I),TO
R(Q+6,I),TOR(Q+7,I),TOR(Q+8,
I)
1160 NEXT Q
1170 NEXT I :: CLOSE #1 :: G
OTO 160
1180 ZEIT=ZEIT-3 :: DISPLAY
AT(14,7):"CA.";ZEIT;"SEKUNDE
N" :: RETURN
1190 CALL KEY(0,K,S):: IF S<
>0 THEN TIP=0 :: GOTO 160 EL
SE 1190
1200 IF TIP THEN 1210 ELSE R
ETURN
1210 IF KL(C)<KL(CC)THEN 124
0
1220 IF (KL(C)-KL(CC))>=9 TH
EN 1260
1230 CT=INT(RND*2):: CTT=INT
(RND*3)+1 :: RETURN
1240 IF (KL(CC)-KL(C))>=9 TH
EN CT=INT(RND*4)+1 :: CTT=IN
T(RND*2):: IF CT=CTT THE
N 1240 ELSE RETURN
1250 CT=INT(RND*3)+1 :: CTT=
INT(RND*2):: RETURN
1260 CT=INT(RND*2):: CTT=INT
(RND*4)+1 :: IF CT=CTT THEN
1260 ELSE RETURN
1270 DISPLAY AT(10,5)ERASE A
LL BEEP:"HABEN SIE VERGESSEN
" :: DISPLAY AT(12,3):"E
RGEBNISSE ZU SPEICHERN?" ::
DISPLAY AT(16,13):"(J/N)"
1280 ACCEPT AT(16,19)VALIDAT
E("JN"):A$ :: IF A$="J" THEN
 1120 ELSE END
1290 REM TOMISLAV RISTIC
1300 REM AUF BOEHLINGSHOF 49
1310 REM 4650 GELSENKIRCHEN
```

Galaxy-Fight

für den TI-99/4A mit Extended Basic

Galaxy-Fight – ein Spiel nach typischer Weltraummanier!
Raumcommander Kirk schickt zwei seiner besten Besatzungsmitglieder in die unerforschten Weiten des Weltalls, mit dem Auftrag, feindliche Klingonenschiffe zu zerstören.
Dabei hilft die "Feuertaste" des TI-99, die hier als Photonenkanone in Aktion tritt.
Wir wünschen viel Spaß!
PS. Eine genaue Anleitung zum Spiel wird gleich nach dem Laden des Programmes gegeben.

```
10 !
20 ! 12/83   (C) BY :
30 ! ***********************
40 ! *                     *
50 ! *     UWE FECHTER     *
60 ! *                     *
70 ! *    JUERGEN FECHTER  *
80 ! *                     *
90 ! *     7965 OSTRACH 1  *
100 !*                     *
110 !* HEILIGENBERGERSTR.6 *
120 !*                     *
130 !***********************
140 !
150 ! VERWENDETE VARIABLEN
160 !
170 ON BREAK NEXT
180 GOTO 220 :: Z,Z1,V,W,W1,V1,W3,W4,V3,V4,T,T1,A,A1,A3,B,B1,C,D,E
,F,G,H,I,J,J1,M,N,O,Q1,P,P1,M1,N1.EIN,ST,X1,X2,Y1,Y2,SI,TR,TR1,TR2
,TR4=TR5
190 TITEL$,A$=Y$
200 CALL CLEAR :: CALL CHAR :: CALL SCREEN :: CALL MOTION :: CALL
PATTERN :: CALL SOUND :: CALL KEY :: CALL HCHAR :: CALL COINC :: C
ALL POSITION
210 CALL DELSPRITE :: CALL COLOR :: CALL SPRITE :: CALL JOYST
220 !@P-
230 !
240 ** ANWEISUNGEN ? **
250 !
260 CALL CLEAR :: CALL SCREEN(7) :: TITEL$="BENOETIGEN2SIE2ANWEISU
NGEN?" :: CALL CHAR(50,"0")
270 CALL COLOR(3,2,12,4,2,12,5,2,12,6,2,12,7,2,12,8,2,12):: FOR I=
1 TO 27 :: CALL HCHAR(12,3+I,ASC(SEG$(TITEL$,I,1))) :: NEXT I
280 CALL KEY(0,EIN,ST) :: IF EIN=74 OR EIN=106 THEN 350 ELSE IF EI
N=78 OR EIN=110 THEN 360 ELSE CALL HCHAR(12,1,32,32)
290 FOR I=1 TO 30 :: NEXT I
300 CALL COLOR(3,2,8,4,2,8,5,2,8,6,2,8,7,2,8,8,2,8) :: FOR I=1 TO
27 :: CALL HCHAR(12,3+I,ASC(SEG$(TITEL$,I,1))) :: NEXT I
310 CALL KEY(0,EIN,ST) :: IF EIN=74 OR EIN=106 THEN 350 ELSE IF EIN
=78 OR EIN=110 THEN 460 ELSE CALL HCHAR(12,1,32,32):: GOTO 270
320 !
330 ! ** SPIELREGELN:  **
340 !
350 CALL CLEAR :: PRINT "G A L A X Y - F I G H T": : :"DER SCHAUPL
ATZ DIESES SPIELS"::"IST DER WELTRAUM, ZWEI"
360 PRINT "SPIELER MANOEVRIEREN MIT DEN": :"JOYSTICKS IHR RAUM- ":
 :"SCHIFF DURCH DAS ALL UND": :"VERSUCH DURCH BETAETIGEN"
370 PRINT "DER FEUERTASTE DEN GEGNER": :"MIT" : : : : : : :"DER PH
OTONENKANONE ZU ZER-" : : : : : : :"STOEREN"
380 GOSUB 2590
390 CALL CLEAR :: PRINT "DIE RAUMSCHIFFE KOENNEN ": :"NICHT UM 180
 GRAD WENDEN, ": :"ABER EIN DIAGONALFLUG IST ": :"MOEGLICH !  "
400 PRINT "FLIEGT EIN RAUMSCHIFF UEBER": :"DEN BILDSCHIRM HINAUS,
SO ": :"TAUCHT ES AUF DER GEGEN-": :"UEBERLIEGENDEN SEITE WIEDER"
410 PRINT :"AUF, DENNOCH IST ES MOEGLICH DEN ": :"BILDSCHIRM SCHEI
NBAR ZU": :"VERLASSEN, INDEM MAN NACH": : :  ::  GOSUB 2590
420 CALL CLEAR ::  PRINT ::"VERLASSEN DES BILDSCHIRMS": :"SOFORT DI
E RICHTUNG WECHSELT": :", SO MAN PARALLEL ZUM ": :"BILDSCHIRMRAND
FLIEGT."
430 PRINT :"JEDOCH IST ER VOR EINEM": :"ANGRIFF NICHT GESCHUETZT U
ND": :"KANN AUCH DORT ABGESCHOSSEN": :"WERDEN" ": :"DAS SPIEL BEGI
NNT NACHDEM"
440 PRINT :"BEIDE JOYSTICKS GLEICHZEITIG": :"BEWEGT WURDEN!": : :
 ::  GOSUB 2590
450 CALL CLEAR :: PRINT "JOYSTICKS ANGESCHLOSSEN? ": :"ALPHA-LOCK-
TASTE GELOEST?": : : : : : : : : : : : :  "NA DANN **** VIEL SPASS
 ***": : : : :: GOSUB 2590
460 CALL CLEAR
470 !
```

```
480 ! ZEICHENDEFINITION
490 !
500 !
510 ! SPIRALNEBEL
520 !
530 CALL CHAR(34,"0C023944821F2747479792929149 0C03402024A4A8C8D2E4
C8F2847800887000")
540 CALL CHAR(128,"0C023944821F2747479792929149 0C03402024A4A8C8D2E
4C8F2847800887000")
550 CALL CHAR(139,"0C023944821F2747479792929149 0C03402024A4A8C8D2E
4C8F2847800887000")
560 !
570 ! PLANETEN
580 !
590 CALL CHAR(43,"03070C101F3F3C1D2733181E0F070300 80C0E0F0F098B8F0
C89830F0E0C08000")
600 CALL CHAR(66,"03070C101F3F3C1D2733181E0F070300 80C0E0F0F098B8F0
C89830F0E0C08000")
610 CALL CHAR(72,"8235744ADEFC4E7A")
620 CALL CHAR(73,"449B2A1F75653F0C")
630 !
640 ! KOMET
650 !
660 CALL CHAR(74,"0000004001071F3F3860408002000400000020000C0E0F8D6
796C320904020120")
670 !
680 ! ASTEROIDEN
690 !
700 CALL CHAR(86,"0048303148002002")
710 CALL CHAR(87,"1092543838549210")
720 !
730 ! RAUMSCHIFF 1 NACH OBEN
740 !
750 CALL CHAR(51,"1010387CFE921038")
760 !
770 ! RAUMSCHIFF 1 NACH RECHTS
780 !
790 CALL CHAR(49,"30189CFF9C183000")
800 !
810 ! RAUMSCHIFF 1 NACH UNTEN
820 !
830 CALL CHAR(50,"381092FE7C381010")
840 !
850 ! RAUMSCHIFF 1 NACH LINKS
860 !
870 CALL CHAR(48,"0C1839FF39180C00")
880 !
890 ! RAUMSCHIFF 1 NACH LINKS OBEN
900 !
910 CALL CHAR(52,"8060703F1D181713")
920 !
930 ! RAUMSCHIFF 1 NACH RECHTS OBEN
940 !
950 CALL CHAR(53,"01060EFCF83848C8")
960 !
970 ! RAUMSCHIFF 1 NACH LINKS UNTEN
980 !
990 CALL CHAR(54,"13121C1F3F706080")
1000 !
1010 ! RAUMSCHIFF 1 NACH RECHTS UNTEN
1020 !
1030 CALL CHAR(55,"C84838F8FC0E0601")
1040 !
1050 ! EXPLOSION DER RAUMSCHIFFE
1060 !
1070 CALL CHAR(136,"1491385CFEB814B1")
1080 CALL CHAR(137,"5492289234501428")
1090 CALL CHAR(138,"8822800441008122")
1100 !
1110 ! SCHUSS
1120 !

1130 CALL CHAR(65,"0000241818240000")
1140 !
1150 ! RAUMSCHIFF 2 NACH OBEN
1160 !
1170 CALL CHAR(59,"183C18183C7EDB99")
1180 !
1190 RAUMSCHIFF 2 NACH RECHTS
1200 !
1210 CALL CHAR(57,"C06032FFFF3260C0")
1220 !
1230 ! RAUMSCHIFF 2 NACH UNTEN
1240 !
1250 CALL CHAR(58,"99DB7E3C18183C18")
1260 !
1270 ! RAUMSCHIFF 2 NACH LINKS
```

```
1280 !
1290 CALL CHAR(56,"03064CFFFF4C0603")
1300 !
1310 ! RAUMSCHIFF 2 NACH LINKS OBEN
1320 !
1330 CALL CHAR(60,"8060703F1D181713")
1340 !
1350 ! RAUMSCHIFF 2 NACH RECHTS OBEN
1360 !
1370 CALL CHAR(61,"01060EFCF83848C8")
1380 !
1390 ! RAUMSCHIFF 2 NACH LINKS UNTEN
1400 !
1410 CALL CHAR(62,"13121C1F3F706080")
1420 !
1430 ! RAUMSCHIFF 2 NACH RECHTS UNTEN
1440 !
1450 CALL CHAR(63,"C84838F8FC0E0601")
1460 !
1470 ! H A U P T P R O G R A M M
1480 !
1490 CALL CLEAR :: FOR I=1 TO 14 :: CALL COLOR(I,2,1) :: NEXT I
1500 RANDOMIZE :: CALL SCREEN(2)
1510 !
1520 ! ZEICHENDEFINITION
1530 !
1540 Y$="020800000000000"
1550 FOR I=80 TO 85
1560 FOR F=1 TO 16 :: J=INT(RND*15+1)
1570 A$=A$&SEG$(Y$,J,1) :: NEXT F
1580 IF A$="0000000000000000" THEN A$="" :: GOTO 1560
1590 CALL CHAR(I,A$) :: A$="" :: NEXT I
1600 !
1610 ! SETZEN DER ASTEROIDEN
1620 !
1630 FOR I=1 TO 16 :: CALL HCHAR(RND*23+1,RND*31+1,80) :: CALL HCHA
R(RND*23+1,RND*31+1,81) :: CALL HCHAR(RND*23+1,RND*31+1,82)
1640 CALL HCHAR(RND*23+1,RND*31+1,83) :: CALL HCHAR(RND*23+1,RND*3
1+1,84) :: CALL HCHAR(RND*23+1,RND*31+1,85) :: CALL HCHAR(RND*23+1
,RND*31+1,86)
1650 NEXT I
1660 FOR I=1 TO 2 :: CALL HCHAR(RND*23+1,RND*31+1,72):: CALL HCHAR(
RND*23+1,RND*31+1,73):: CALL HCHAR(RND*23+1,RND*31+1,87):: NEXT I
1670 !
1680 ! SETZEN DER PLANETEN, KOMETEN UND DER SPIRALNEBEL
1700 E=74 :: GOSUB 2810 :: GOSUB 2810 :: E=34 :: GOSUB 2810 :: E=
128 :: GOSUB 2810 :: E=139 :: E=43 :: GOSUB 2810 :: GOSUB 2810 ::
 E=66 :: GOSUB 2810
1710 CALL COLOR(1,4,1,13,10,1,14,14,1,3,4,1,4,5,1,6,12,1,7,16,1,2,
13,1,5,6,1)
1720 X1=(INT(RND*20)+2)*8
1730 X2=(INT(RND*20)+2)*8
1740 Y1=(INT(RND*28)+2)*8
1750 Y2=(INT(RND*28)+2)*8 :: I=0 :: TR=RND*7+48 :: TR1=RND*7+56
1760 CALL SPRITE(#1,TR,4,X1,Y1,0,0,#2,TR1,8,X2,Y2,0,0)
1770 CALL JOYST(1,V,W):: CALL JOYST(2,V3,W3) :: CALL KEY(1,Q,Q1) :

: CALL KEY(2,P,P1):: T=ABS(V)+ABS(W) :: T1=ABS(V3)+ABS(W3)
1780 IF T1=0 OR T2=0 THEN CALL DELSPRITE(#1,#2):: GOTO 1760 ELSE 1
820
1790 I,Q,P=0
1800 CALL JOYST(1,V,W):: CALL KEY(1,Q,Q1)
1810 CALL JOYST(2,V3,W3):: CALL KEY(2,P,P1)
1820 M=(2*V+5*W+W)/8+5 :: N=(2*V3+5*W3+W3)/8+5 :: IF M=Z THEN 192
0
1830 ON M GOTO 1900,1860,1910,1840,1920,1850,1880,1870,1890
1840 A=0:: B=-20:: Z=6:: C=48:: GOTO 1920
1850 A=0:: B=20:: Z=4:: C=49:: GOTO 1920
1860 A=20:: B=0:: Z=8:: C=50:: GOTO 1920
1870 A=-20:: B=0:: Z=2:: C=51:: GOTO 1920
1880 A=-15:: B=-15:: Z=3:: C=52:: GOTO 1920
1890 A=-15:: B=15:: Z=1:: C=53:: GOTO 1920
1900 A=15:: B=-15:: Z=9:: C=54:: GOTO 1920
1910 A=15:: B=15:: Z=7:: C=55
1920 IF N=Z1 THEN 2020
1930 ON N GOTO 2000,1960,2010,1940,2020,1950,1980,1970,1990
1940 A1=0:: B1=-20:: Z1=6:: D=56:: GOTO 2020
1950 A1=0:: B1=20:: Z1=4:: D=57:: GOTO 2020
1960 A1=20:: B1=0:: Z1=8:: D=58:: GOTO 2020
1970 A1=-20:: B1=0:: Z1=2:: D=59:: GOTO 2020
1980 A1=-15:: B1=-15:: Z1=3:: D=60:: GOTO 2020
1990 A1=-15:: B1=15:: Z1=1:: D=61:: GOTO 2020
2000 A1=15:: B1=-15:: Z1=9:: D=62:: GOTO 2020
2010 A1=15:: B1=15:: Z1=7:: D=63
2020 CALL PATTERN(#1,C,#2,D):: CALL MOTION(#1,A,B,#2,A1,B1)
2030 IF Q+1 OR P+1 THEN 2080
2040 GOTO 1800
```

```
2050 !
2060 ! SCHUSS ABGEBEN UND UEBERPRUEFEN
2070 !
2080 IF Q+1 AND P+1 THEN CALL POSITION(#1,E,F,#2,C,H):: CALL SPRIT
E(#3,42,14,E,F,6*A,6*B,#4,65,10,C,H,6*A1,6*B1):: GOTO 2160
2090 IF Q+1 THEN CALL POSITION(#1,E,F):: CALL SPRITE(#3,42,14,E,F,
6*A,6*B):: GOTO 2130 ELSE CALL POSITION(#2,C,H):: CALL SPRITE(#4,6
5,10,C,H,6*A1,6*B1)
2100 CALL SOUND(40,-7,0)
2110 CALL COINC(#4,#1,10,TR5):: IF TR5 THEN 2320
2120 I=I+1:: IF I<3 THEN 2100 ELSE CALL DELSPRITE(#4):: GOTO 1790
2130 CALL SOUND(40,-6,0)
2140 CALL COINC(#3,#2,10,TR4):: IF TR4 THEN 2280
2150 I=I+1:: IF I<3 THEN 2130 ELSE CALL DELSPRITE(#3):: GOTO 1790
2160 CALL SOUND(50,-5,0)
2170 CALL COINC(#3,#4,8,TR4)
2180 CALL COINC(#2,#3,10,TR1):: CALL COINC(#4,#1,10,TR2):: IF TR1
OR TR OR TR2 THEN 2200
2190 I=I+1:: IF I<2 THEN 2170 ELSE CALL DELSPRITE(#3,#4):: GOTO 17
90
2200 IF TR=0 THEN 2240 ELSE CALL MOTION(#3,-6*A,-6*B):: CALL PATTE
RN(#4,32,#3,32):: CALL DELSPRITE(#4)
2210 FOR I=1 TO 10 :: NEXT I
2220 CALL PATTERN(#3,138):: CALL MOTION(#3,0,0):: CALL SOUND(300,-
6,0):: FOR I=1 TO 50 :: NEXT I
2230 CALL DELSPRITE(#3):: GOTO 1790
2240 IF TR1 AND TR2 THEN 2480 ELSE IF TR1 THEN 2280 ELSE 2320
2250 !
2260 ! SIEGER RAUMSCHIFF 1
2270 !
2280 SI=49 :: CALL MOTION(#2,5,0):: A3=2 :: GOTO 2330
2290 !
2300 ! SIEGER RAUMSCHIFF 2
2310 !
2320 SI=56 :: CALL MOTION(#1,5,0):: A3=1
2330 CALL DELSPRITE(#3,#4):: FOR I=136 TO 138
2340 J1=J1+1 :: CALL SOUND(-600,-8+J1,0)
2350 CALL PATTERN(#A3,I)
2360 FOR J=1 TO 60 :: NEXT J
2370 NEXT I
2380 FOR I=1 TO 700:: NEXT I

2390 CALL CLEAR :: CALL DELSPRITE(ALL) :: CALL SCREEN(15)
2400 DISPLAY AT(10,2): "pilot mit ";CHR$(SI);"ist sieger !"
2410 GOSUB 2620 :: DISPLAY AT(13,5): "noch ein spiel !"
2420 ACCEPT AT(13,21)VALIDATE("NnjJ") SIZE(-1) BEEP: Y$
2430 IF Y$="J" OR Y$="j" THEN RUN 1490
2440 END
2450 !
2460 ! KEIN SIEGER
2470 !
2480 CALL DELSPRITE(#3,#4):: CALL MOTION(#1,5,0,#2,5,0)
2490 FOR I=136 TO 138 :: NEXT I
2500 J1=J1+1 :: CALL SOUND(-600,-8+J1,0)
2510 CALL PATTERN(#1,I,#2,I)
2520 FOR J=1 TO 60 :: NEXT J
2530 NEXT I
2540 FOR I=1 TO 700 :: NEXT I
2550 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL SCREEN(15)
2560 DISPLAY AT(8,2): "beide piloten verloren ihr"
2570 DISPLAY AT(10,13): "leben !"
2580 GOTO 2410
2590 DISPLAY AT(24,25): ">>"
2600 CALL KEY(0,EIN,ST)
2610 IF ST=0 THEN 2600 ELSE RETURN
2620 CALL SOUND(150,349,5)
2630 CALL SOUND(150,449,5)
2640 CALL SOUND(150,523,5)
2650 CALL SOUND(150,659,5)
2660 CALL SOUND(300,494,5,784,5)
2670 CALL SOUND(150,659,5)
2680 CALL SOUND(150,494,5,698,5,784,5)
2690 CALL SOUND(145,494,5,698,5,784,5)
2700 CALL SOUND(150,659,30)
2710 CALL SOUND(150,659,5)
2720 CALL SOUND(145,466,5,659,5,784,5)
2730 CALL SOUND(145,466,5,659,5,784,5)
2740 CALL SOUND(150,659,30)
2750 CALL SOUND(150,659,5)
2760 CALL SOUND(295,494,5,698,5,784,5)
2770 CALL SOUND(145,370,0,523,0,622,0)
2780 CALL SOUND(295,349,0,494,0,587,0)
2790 CALL SOUND(800,330,0,392,0,523,0)
2800 RETURN
2810 F=INT(RND*22)+1 :: G=INT(RND*30)+1
2820 CALL HCHAR(F,G,E):: CALL HCHAR(F+1,G,E+1):: CALL HCHAR(F,G+1,
E+2):: CALL HCHAR(F+1,G+1,E+3):: RETURN
```

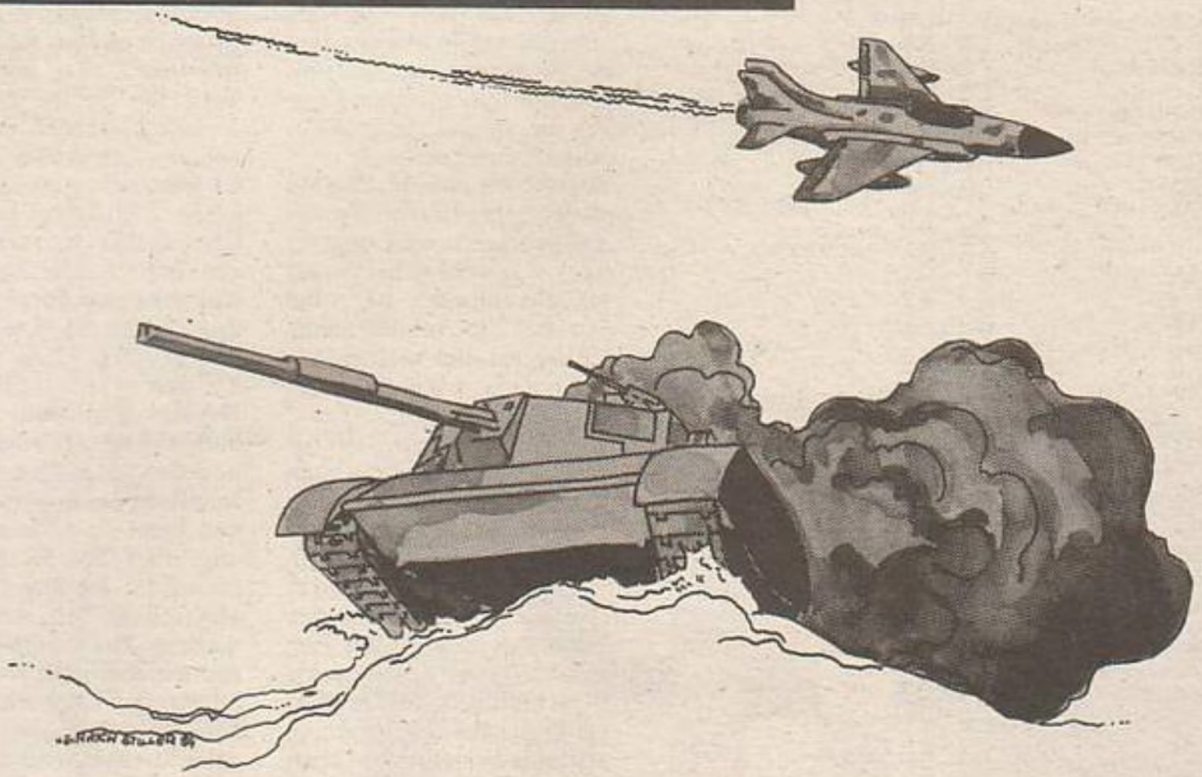
KORREKTUR

Das Listing zum Programm Würgmann, aus HC 8/84, muß um folgende Zeilen ergänzt werden:

```
3200 REM LABYRINTHMAUERN PER ZUFALLSGENERATORE AENDERN UND ZUGEHOERIGE VARIABLEN
 SETZEN
3210 REM
3220 IF A2(2)=1 THEN 3310
3230 REM
3240 REM WAAGRECHT
3250 REM
3260 LET C1=A2(2)
3270 LET C2=A2(4)
3280 CALL CHAR(103+C2,A2$(C1))
3290 LET A7(C2)=C1
3300 GOTO 2470
3310 REM
3320 REM SENKRECHT
3330 REM
3340 LET C1=A2(2)
3350 LET C2=A2(4)
3360 CALL CHAR(107+C2,A2$(C1+2))
3370 LET A7(C2+4)=C1
3380 GOTO 2470
3390 REM
3400 REM PUNKTE FRESSEN UND LOESCHEN - SCORE ERHOEHEN - SOUND ERZEUGEN
3410 REM
3420 REM PUNKTE FRESSEN
3430 REM CHR$(136)
3440 IF A5<>136 THEN 3590
3450 CALL HCHAR(B8,B9,C5)
3460 CALL HCHAR(A0,A1,A6)
3470 LET C3=C3+1
3480 LET A8=A8+70
3490 LET A5=13
3500 LET C5=32
3510 CALL SOUND(220,523,0,659,0)
3520 CALL SOUND(220,587,0,698,0)
3530 CALL SOUND(220,659,0,784,0)
3540 CALL SOUND(220,587,0,698,0)
3550 CALL SOUND(220,523,0,659,0)
3560 GOSUB 4450
3570 IF C3=12 THEN 4150
3580 RETURN
3590 REM CHR$(144)
3600 IF A5<>144 THEN 3750
3610 CALL HCHAR(B8,B9,C5)
3620 CALL HCHAR(A0,A1,A6)
3630 LET C3=C3+1
3640 LET A8=A8+60
3650 LET A5=13
3660 LET C5=32
3670 CALL SOUND(170,262,0,523,3,1047,2)
3680 CALL SOUND(170,330,0,659,3,1319,2)
3690 CALL SOUND(170,349,0,698,3,1397,2)
3700 CALL SOUND(170,330,0,659,3,1319,2)
3710 CALL SOUND(170,262,0,523,3,1047,2)
3720 GOSUB 4450
3730 IF C3=12 THEN 4150
3740 RETURN
3750 REM CHR$(130)
3760 IF A5<>130 THEN 3920
3770 CALL HCHAR(B8,B9,C5)
3780 CALL HCHAR(A0,A1,A6)
3790 LET C3=C3+1
3800 LET A8=A8+20
3810 LET A5=13
3820 LET C5=32
3830 FOR C9=1 TO 3
3840 CALL SOUND(190,220,0,440,4,880,2)
3850 NEXT C9
3860 CALL SOUND(190,196,0,392,4,784,2)
3870 CALL SOUND(190,262,0,523,4,1047,2)
3880 CALL SOUND(190,220,0,440,4,880,2)
3890 GOSUB 4450
3900 IF C3=12 THEN 4150
3910 RETURN
3920 REM CHR$(152)
3930 IF A5<>152 THEN 4070
3940 CALL HCHAR(B8,B9,C5)
3950 CALL HCHAR(A0,A1,A6)
3960 LET C3=C3+1
3970 LET A8=A8+50
3980 LET A5=13
3990 LET C5=32
4000 CALL SOUND(210,262,0,523,5,1047,1)
4010 CALL SOUND(210,175,0,349,5,698,1)
4020 CALL SOUND(210,247,0,494,5,988,1)
4030 CALL SOUND(210,220,0,440,5,880,1)
4040 CALL SOUND(210,196,0,392,5,784,1)
4050 GOSUB 4450
4060 IF C3=12 THEN 4150
4070 RETURN
4080 IF A8>A9 THEN 4100
4090 RETURN
4100 A9=A8
4110 FOR A4=1 TO LEN(STR$(A9))
4120 CALL HCHAR(22,18-LEN(STR$(A9))+A4,ASC(SEG$(STR$(A9),A4,1)))
4130 NEXT A4
4140 RETURN
4150 REM
4160 REM NAECHSTE RUNDE
4170 REM
4180 CALL SOUND(200,523,4,1047,2)
4190 CALL SOUND(200,784,4,1568,2)
4200 CALL SOUND(200,780,4,1480,2)
4210 CALL SOUND(200,784,4,1568,2)
4220 CALL SOUND(200,740,4,1580,2)
4230 CALL HCHAR(22,27,48,3)
4240 FOR A4=1 TO 2
4250 CALL SOUND(200,523,4,784,4,1047,2)
4260 NEXT A4
4270 IF A8<A9 THEN 4290
4280 A9=A8
4290 LET C8=C8-10
4300 IF C8<10 THEN 4310 ELSE 4320
4310 LET C8=10.1
4320 LET A8=A8+INT(A3+1)*10
4330 LET A3=C8
4340 LET C3=0
4350 CALL HCHAR(A0,A1,C5)
4360 LET A0=0
4370 LET A1=0
4380 GOSUB 4450
4390 GOSUB 2020
4400 FOR A4=104 TO 107
4410 CALL CHAR(A4,"0000FF0000FF0000")
4420 CALL CHAR(A4+4,"2424242424242424")
4430 NEXT A4
4440 GOTO 2280
4450 FOR A4=1 TO LEN(STR$(A8))
4460 CALL HCHAR(23,18-LEN(STR$(A8))+A4,ASC(SEG$(STR$(A8),A4,1)))
4470 NEXT A4
4480 RETURN
4490 GOSUB 4080
4500 FOR A4=1 TO 2
4510 FOR A3=0 TO 4
4520 CALL SOUND(-100,587,A3)
4530 NEXT A3
4540 FOR A3=5 TO 9
4550 CALL SOUND(-100,659,A3)
4560 NEXT A3
4570 FOR A3=10 TO 14
4580 CALL SOUND(-100,523,A3)
4590 NEXT A3
4600 FOR A3=15 TO 19
4610 CALL SOUND(-100,587,A3)
4620 NEXT A3
4630 FOR A3=20 TO 28
4640 CALL SOUND(-100,494,A3)
4650 NEXT A3
4660 NEXT A4
4670 IF A8>A9 THEN 4680 ELSE 4700
4680 LET A9=A8
4690 GOSUB 4080
4700 PRINT : :"    'eeee
```

```
4770 NEXT A3
4780 CALL KEY(0,B1,B2)
4790 IF B2 THEN 4800 ELSE 4780
4800 CALL CLEAR
4810 GOTO 1060
4820 CALL CLEAR
4830 PRINT "   'eeee
```

Fighter-Command
für den TI-99/4A Extended Basic



Stellen Sie Ihren Monitor oder Fernseher nochmals ein, damit Sie den guten Ton und die Grafikeffekte von Fighter-Command genießen können.

Sie steuern einen Jagdbomber, der gegen Panzer, Abfangjäger und Schiffe kämpfen muß.

Phase 1:
Angriff der Panzer: Zwei Panzer, ausgerüstet mit Fliegerabwehrkanonen, greifen Sie an. Ihr Jagdbomber (mit Senkrechtstarteigenschaften) kann sich dagegen wehren, indem Sie Bomben auf die Panzer werfen, die durch Drücken des Feuerknopfes ausgelöst werden.
Für einen getroffenen Panzer erhalten Sie 100 Punkte. Treffen die Panzer aber Sie, dann wird Ihnen eines Ihrer 5 Flugzeuge abgezogen.
Der Senkrechtstarter bewegt sich auf und ab (Joystick). Sie können das Flugzeug beschleunigen (Joystick nach rechts) oder abbremsen bzw. rückwärts fliegen (Joystick nach links).
In den Boden dürfen Sie Ihr Flugzeug natürlich nicht rammen. Wenn Sie 1100 Punkte erreicht haben, geht's weiter im Programm.

Phase 2:
Nun wird's schwieriger. Drei Abfangjäger greifen an, die sich ebenfalls wie Sie im Tiefflug befinden. Mit feuerspeienden Kanonen kommen sie Ihnen entgegen.
Aber Sie sollen sich ja nicht gleich abknallen lassen. Schießen Sie den Jägern einfach eine Ladung Blei aus Ihren Maschinenkanonen entgegen (durch Drücken des Feuerknopfes). Wenn Sie 10 Abschüße gemacht haben, geht es weiter in Phase 3.

Phase 3:
Nun befinden Sie sich über dem offenen Meer. Zwei kleine Schnellboote (noch groß genug um tödliche Flag zu tragen) beschießen Sie. Sie setzen wieder Bomben ein um nach 10 versenkten Schiffen zur nächsten Aktion zu gehen.

Phase 4:
Phase 4 ist wie Phase 2, nur die Marineflieger haben diesmal etwas gegen Sie.
Anmerkung: Es gibt Blindgänger in der Munition beider Seiten.

Variablenliste:
A: Variable für READ (Sound)-Schleife
P: Punkte
R: Flugzeuge
HS: Highscore
I,Y,C: Laufvariablen
X,Y: Joystick
GUT: Variable zur Aktionsansteuerung (wenn bestimmte Punktzahl erreicht nächste Phase)
X1,Y1: Spritepositionen
AA,BB,BUM,C: Diverse Treffervariablen
A1,B1,A2,B2 usw.: Spritepositionen des Gegners
COL: Farbbestimmung
A: Zähler für Aktionsdurchgang
K: Tastatur, Feuerknopfanzeige
S: Statusvariable

```
100 !  FIGHTER-COMMAND
      ***************
  (C) STEFAN SCHAPER
110 GUT=1000
120 CALL CHAR(96,"3C4291A1A1
91423C")
130 CALL CHAR(36,"0104200068
4402010102446800240000C04600B
480BCAEC6C0A690819404401
0")
140 CALL CHAR(40,"0104200068
440201",41,"1F3FEF80EFA5EF",
42,"57BA147B8989999D",43
,"FFFFFF01FE55FE")
150 CALL CLEAR :: DISPLAY AT
(15,3):"F I G H T E R  C O M
 M A N D "
160 DISPLAY AT(3,3):"' STEFA
N SCHAPER"
170 FOR I=1 TO 35 :: READ A
:: CALL SOUND(100,A,0):: NEX
T I
180 CALL SOUND(600,1000,1)
190 DISPLAY AT(20,3):"PRESS
FIRE TO START !" :: CALL KEY
(1,K,S):: IF K<>18 THEN
190
200 R=5 :: P=0 :: GUT=1000
210 RANDOMIZE
220 CALL CLEAR :: CALL SCREE
N(3)
230 CALL CHAR(140,"00B1C3E7F
FFFFFFF")
240 CALL HCHAR(24,1,140,32)
250 CALL MAGNIFY(3):: CALL C
HAR(120,"002030283C7F3F1F",1
21,"1F",122,"00000000609
04CFF",123,"FC")
260 CALL CHAR(124,"000000000
000E600",125,"",126,"0000000
0000066",127,"")
270 CALL CHAR(128,"000000402
0100B0F^",129,"1F3FEF80EFA5EF
",130,"000000000000FCFF"
,131,"FFFFFF01FE55FE")
280 CALL CHAR(132,"00B0C0FEF
FFEC080",133,"",134,"",135,"
")
290 CALL CHAR(136,"000000B04
02010",137,"",138,"",139,"")
300 COL=2
310 CALL SPRITE(#1,120,13,90
,40)
320 CALL SPRITE(#10,128,COL,
173,200,0,-20,#11,128,COL,17
3,110,0,-20)
330 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,Y*-4,0)
340 IF X=4 THEN W=W+-5 ELSE
IF X=-4 THEN W=W+5 :: IF W<-
90 THEN W=-90
350 IF W>30 THEN W=30
360 IF W<-100 THEN W=-100
370 CALL MOTION(#10,0,W,#11,
0,W)
380 CALL KEY(1,K,S):: IF K=1
8 THEN GOSUB 500 ELSE CALL P
OSITION(#1,X1,Y1)
390 IF INT(RND*2)=1 THEN GOS
UB 580 ELSE 400
400 CALL POSITION(#12,AA,BB)
:: IF AA<10 THEN CALL DELSPR
ITE(#12,#13)
410 FOR I=10 TO 13 :: CALL C
OINC(#1,#I,16,Q):: IF Q THEN
 650
420 NEXT I
430 CALL POSITION(#1,X1,Y1):
: IF X1>173 THEN 650
440 CALL POSITION(#10,X2,Y2)
:: IF Y2<25 THEN CALL LOCATE
(#10,173,240)
450 CALL POSITION(#11,X3,Y3)
:: IF Y3<25 THEN CALL LOCATE
(#11,173,240)
460 CALL COINC(#10,#3,25,A):
: IF A THEN 740
470 CALL COINC(#11,#3,25,B):
: IF B THEN 790
480 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,Y*-4,0):: CALL KE
Y(1,K,S):: IF K=18 THEN
GOSUB 500
490 GOTO 330
500 CALL POSITION(#1,X1,Y1):
: CALL SOUND(-500,-5,0)
510 CALL SPRITE(#3,132,2,X1,
Y1,30,-10)
520 CALL POSITION(#3,AA,BB):
: FOR I=AA TO 173 STEP 36 ::
 CALL COINC(#10,#3,19,S)
:: CALL COINC(#11,#3,19,SS):
: IF S THEN 740 ELSE IF SS T
HEN 790
530 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,Y*-3,0)
540 FOR C=10 TO 13 :: CALL C
OINC(#C,#1,15,Q):: IF Q THEN
 650
550 NEXT C
560 NEXT I :: CALL DELSPRITE
(#3):: FOR I=1 TO 4 :: NEXT
I :: CALL DELSPRITE(#12,
#13)
570 RETURN
580 IF INT(RND*2)=0 THEN RET
URN ELSE 590
590 CALL POSITION(#10,Z,Q,#1
1,Z1,Q1):: CALL SOUND(-500,-
7,0)
600 CALL SPRITE(#12,136,2,Z+
1,Q,-10,-20,#13,136,2,Z1+1,Q
1,-10,-20)
610 FOR I=1 TO 120 :: FOR C=
10 TO 13 :: CALL COINC(#1,#I
,15,A):: IF A THEN 490 :
: CALL JOYST(1,X,Y):: CALL M
OTION(#1,Y*-3,0)
620 CALL KEY(1,K,S):: IF K=1
8 THEN GOSUB 500
630 NEXT C :: NEXT I :: CALL
 DELSPRITE(#12,#13)
640 GOTO 330
650 CALL SOUND(1000,-7,0)::
CALL PATTERN(#1,36):: FOR I=
1 TO 60 :: NEXT I :: CAL
L DELSPRITE(#1)
660 CALL DELSPRITE(#3)
670 R=R-1 :: IF R=0 THEN 700
 ELSE 680
680 DISPLAY AT(2,3):"SHIPS";
R :: CALL PATTERN(#1,120)::
CALL LOCATE(#1,90,40)
690 GOTO 330
700 CALL DELSPRITE(ALL):: DI
SPLAY AT(15,7):"GAME OVER"
710 DISPLAY AT(20,3):"NEW GA
ME Y OR N ?"
720 CALL KEY(0,K,S):: IF S=0
 THEN 720 ELSE 730
730 IF K=121 THEN 200 ELSE S
TOP
740 CALL SOUND(500,-7,0)
750 CALL PATTERN(#10,40):: F
OR I=1 TO 20 :: NEXT I
760 CALL DELSPRITE(#10,#3)::
 P=P+100 :: DISPLAY AT(1,3):
"SCORE:";P :: IF P>GUT T
HEN 890
770 CALL PATTERN(#10,128)::
CALL LOCATE(#10,173,220):: C
ALL MOTION(#10,0,W)
780 GOTO 330
790 CALL SOUND(500,-7,0):: C
ALL PATTERN(#11,40):: FOR I=
1 TO 20 :: NEXT I :: CAL
L DELSPRITE(#11,#3)
800 P=P+100 :: DISPLAY AT(1,
3):"SCORE:";P :: IF P>GUT TH
EN 890
810 CALL PATTERN(#11,128)::
CALL LOCATE(#11,173,80):: CA
LL MOTION(#11,0,W)
820 GOTO 330
830 DATA 500,440,294,1100,39
3,310,200,800,900,1000,1200,
1000
840 DATA 500,800,874,190,700
,876,330,123,560,827,2200,12
00,350,210
850 DATA 454,294,1100,494,11
10
860 DATA 120,900,610,800
870 CALL MOTION(#10,0,W,#11,
0,W)
880 GOTO 32767
890 CALL MAGNIFY(3):: CALL D
ELSPRITE(ALL)
900 GUT=GUT+1000
910 RESTORE :: FOR I=1 TO 35
 :: READ A :: CALL SOUND(100
,A,0):: NEXT I
920 CALL LOCATE(#1,90,40)
930 CALL CHAR(96,"000000000E
113FFF",97,"3F",98,"0000000B
18387BFEFF",99,"FE")
940 CALL SPRITE(#10,96,2,90,
190,#11,96,2,120,190,#12,96,
2,60,190)
950 CALL SPRITE(#1,120,13,90
,40)
960 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,Y*-3,0)
970 CALL POSITION(#1,X1,Y1):
: IF X1>173 THEN 1330
980 IF X=4 THEN W=W+-5 ELSE
IF X=-4 THEN W=W+5 :: IF W<-
90 THEN W=-90
990 IF W>30 THEN W=30
1000 IF W<-100 THEN W=-100
1010 CALL MOTION(#10,0,W,#11
,0,W,#12,0,W)
1020 IF INT(RND*3)=0 THEN 12
50
1030 CALL KEY(1,K,S):: IF K=
18 THEN GOSUB 1070
1040 FOR I=10 TO 15 :: CALL
COINC(#1,#I,15,Q):: IF Q THE
N 1330
1050 NEXT I
1060 GOTO 960
1070 CALL SOUND(-400,-5,0)::
 CALL POSITION(#1,X,Y)
1080 CALL SPRITE(#3,124,16,X
,Y,0,80)
1090 FOR I=0 TO 3 :: CALL CO
INC(#3,#10,10,A):: IF A THEN
 1130 :: CALL COINC(#3,#
11,10,B):: IF B THEN 1170 ::
 CALL COINC(#3,#12,10,C):: I
F C THEN 1210
1100 FOR C=10 TO 15 :: CALL
COINC(#1,#C,13,Q):: IF Q THE
N 1330
1110 NEXT C
1120 NEXT I :: CALL DELSPRIT
E(#3):: GOTO 960
1130 CALL PATTERN(#10,36)::
CALL SOUND(500,-7,0):: P=P+1
00 :: DISPLAY AT(1,3):"S
CORE";P
1140 IF P>GUT THEN 1380
1150 CALL DELSPRITE(#10,#3):
: CALL PATTERN(#10,96):: CAL
L LOCATE(#10,INT(RND*150
)+1,250)
1160 GOTO 960
1170 CALL PATTERN(#11,36)::
CALL SOUND(500,-7,0):: P=P+1
00 :: DISPLAY AT(1,3):"S
CORE";P
1180 IF P>GUT THEN 1380
1190 FOR I=1 TO 30 :: NEXT I
 :: CALL DELSPRITE(#11,#3)::
 CALL PATTERN(#11,96)::
CALL LOCATE(#11,INT(RND*150)
+1,250)
1200 GOTO 960
1210 CALL PATTERN(#12,36)::
CALL SOUND(500,-7,0):: P=P+1
00 :: DISPLAY AT(1,3):"S
CORE";P
1220 IF P>GUT THEN 1380
1230 FOR I=1 TO 20 :: NEXT I
 :: CALL PATTERN(#12,96):: C
ALL DELSPRITE(#12,#3)::
CALL LOCATE(#12,INT(RND*150)
+1,250)
1240 GOTO 960
1250 FOR I=10 TO 12 :: CALL
POSITION(#I,A1,B1):: CALL SO
UND(400,-7,0)
1260 FOR Y=13 TO 15 :: CALL
SPRITE(#Y,124,2,A1,B1,0,-60)
:: NEXT Y
1270 CALL JOYST(1,X,Y):: CAL
L MOTION(#1,Y*-3,0):: CALL K
EY(1,K,S):: IF K=18 THEN
 GOSUB 1070
1280 NEXT I
1290 FOR I=10 TO 15 :: CALL
COINC(#I,#1,15,Q):: IF Q THE
N 1330 :: CALL JOYST(1,X
,Y):: CALL MOTION(#1,Y*-3,0)
:: CALL KEY(1,K,S)
1300 IF K=18 THEN GOSUB 1070
1310 NEXT I :: FOR I=13 TO 1
5 :: CALL DELSPRITE(#I):: NE
XT I
1320 GOTO 960
1330 FOR I=13 TO 15 :: CALL
DELSPRITE(#I,#3):: NEXT I
1331 CALL SOUND(500,-7,0)
1332 CALL PATTERN(#1,36):: F
OR I=1 TO 30 :: NEXT I :: CA
LL PATTERN(#1,120):: R=R
-1
1333 IF R=0 THEN 1360
1340 DISPLAY AT(2,3):"SHIPS"
;R :: FOR I=13 TO 15 :: CALL
 DELSPRITE(#1,#I):: NEXT
 I
1350 CALL PATTERN(#1,120)::
CALL LOCATE(#1,90,40):: GOTO
 960
1360 CALL DELSPRITE(ALL):: D
ISPLAY AT(13,3):"GAME OVER"
:: DISPLAY AT(20,3):"NEW
 GAME Y OR N"
1370 CALL KEY(0,K,S):: IF S=
0 THEN 1370 :: IF K=121 THEN
 200 ELSE STOP
1380 CALL CLEAR :: CALL SCRE
EN(8):: CALL CHAR(140,"0060B
018FFFFFFFF"):: CALL HCH
AR(24,1,140,32)
1390 CALL CHAR(128,"",129,"4
02D1FFF7F3F1F00",130,"",131,
"82DCDBFFFEFCFB00")
1400 CALL CHAR(40,"002000423
1280502",41,"402D1FFF7F3F100
",42,"002008410628 1020",
43,"82DCDBFFFEFCFB00")
1410 A=A+1 :: IF A/2<>1 OR A
/2<>2 THEN 310 ELSE 220 :: I
F A/2<>1 THEN COL=15 ELS
E COL=2
1420 GOTO 310
```

für den TI-99/4A Extended Basic

HERC UND ORAKEL VO

Beim Eintippen des recht umfangreichen Listings, welches den Arbeitsspeicher des TI-99/4A nahezu komplettiert, kann es bei einigen Zeilen vorkommen, daß der Cursor "ansteht". In diesem Fall einfach "Enter" drücken, obwohl die betreffende Zeile noch unvollständig ist, eine etwaige Fehlermeldung bitte ignorieren, anschließend "FCTN" und "REDO" drücken und so die Zeile beenden.

Nach dem Start mit "RUN" (Alphalock-Taste lösen) erscheint der Bildschirmaufbau und der gewünschte Schwierigkeitsgrad ist mit "Enter" einzugeben. 3 Schwierigkeitsstufen sind möglich:

1: ist zum "Kennenlernen" und ganz einfach zu gewinnen, da Herkules genug Geld hat.

2: auch noch einfach

3: man muß schon aufpassen, um nicht vorzeitig pleite zu gehen.

Danach den Feuerknopf der Fernbedienung drücken um das Spiel zu starten. Kurze Zeit nach Eingang erscheinen 3 Tempel, Herkules und jede Menge schrecklicher Ungeheuer: 2 grausige Spinnen, 4 Krokodile schwimmen im Fluß der Unterwelt namens "Acheron", das jenseitige Ufer wird von einem feuerspeienden Drachen bewacht und die Tempel vom 3-köpfigen Höllenhund "Zerberus".

1) Herkules:

Wird mit Fernbedienung 1 (Joystick 1) gesteuert, jede Richtung ist möglich.

Ziel des Spieles ist, unter 3 Tempeln, jenen des "Hades" zu zerstören, um zum "goldenen Vlies" zu gelangen. Die Zerstörung des jeweiligen Tempels tritt ("computergesteuert") ein, wenn sich Herkules genau unter den Torbogen stellt. Wie seinerzeit in den legendären "Herkules-Filmen", im Kino, stürzt der betreffende Tempel mit Getöse ein. Wurde der richtige Tempel zerstört, liegt das goldene Vlies Herkules zu Füßen und ein Lied erklingt.

Wurde aber versehentlich (man muß es ja erraten) der falsche Tempel zerstört, gibt es 500 Drachmen Abzug vom Reisespesenkonto des Herkules und er wird wieder an seinen Ausgangspunkt zurückversetzt. Das Spiel ist aus, wenn Herkules blank ist oder er wird von einem der zahlreichen Ungeheuer erwischt. Im ersten Fall wird er von Zeus zur Strafe, weil er alles verpraßt hat in einen schäbigen Pleitegeier verwandelt, welcher in der oberen Bildschirmmitte nach einigen hilflosen Flügelschlägen sitzen bleibt.

Im zweiten Fall wird Herkules von dem entsprechenden Ungeheuer, in jeweils eigener Manier, beseitigt. Die Spinne überfällt ihn, das Krokodil frißt ihn ganz einfach auf, der Drachen bespeit Herkules ausgiebigst mit Feuer und Zerberus, der dreiköpfige Höllenhund, beißt ihn.

Da nach dem Start ein Durchkommen zu den 3 Tempeln völlig unmöglich ist, muß sich Herkules des "Orakels von Delphi" bedienen.

2) Das Orakel von Delphi:

Durch Drücken des Feuer-Knopfes (Act.-Taste) erscheint am unteren Bildschirmrand ein Orakelspruch, welcher den Ablauf des Spieles (Geschwindigkeit der Sprites) wesentlich beeinflußt. Anschließend wird eine meist unverschämte "Orakelgebühr" in Rechnung gestellt, welche vom Reisespesenkonto des Herkules abgezogen wird, dann werden zwei Gegenstände oder Waffen ausgegeben, welche Herkules zur Beseitigung der Ungeheuer verwenden kann und der Saldo des restlichen Geldes wird angezeigt.

Aber das Orakel ist meist zweideutig, launisch und schadenfroh, denn wenn Herkules von einem Ungeheuer erwischt wird, stellt dies das Orakel boshafter Weise auch noch schriftlich fest.

Ist das Orakel übel gelaunt, kommt es vor, daß der freche Orakelspruch: "höre meine Lehren, nun kannst Du Dich zum Hades scheren", erscheint und gemeiner Weise werden keine Waffen ausgegeben aber dafür 70 Drachmen "Orakelgebühr" verrechnet. Es kommt aber auch vor, daß das Orakel gut aufgelegt ist und gratis Waffen ausgibt.

Mit den beiden ausgegebenen Gegenständen (Faß, Speer, Amphore, Axt oder Felsbrocken) kann man die Ungeheuer bewerfen und dieselben gegebenenfalls zum Verschwinden bringen.

ULES

DAS
ON DELPHI

um möglichst unbehelligt zu den Tempeln zu gelangen.
Allerdings darf man, je nach dem gewählten Schwierigkeitsgrad, das Orakel nicht zu oft befragen um wieder neue Waffen zu erhalten, da das Orakel gebührenpflichtig und Herkules pleite ist, ist auch das Spiel zu Ende.

3) Die Fernbedienung hat mehrere Funktionen:
Feuerknopf (Act.-Taste):
Start des Spieles am Anfang und Ende.
Während des Spieles wird das Orakel befragt um auch Waffen zu erhalten.
Joystick:
Steuerung für Herkules in jede beliebige Richtung. Achtung! Wenn Herkules durch den "Acheron" schwimmt ist zu beachten, daß er wesentlich langsamer schwimmen als laufen kann. Es ragt zwar nur sein Oberkörper aus dem Wasser aber die Krokodile sind sehr gefährlich.

Steuerung der Richtung für den Wurf und Auslösung:
Es sind nur drei Richtungen möglich: Senkrecht nach oben, schräg nach links oder rechts (45 Grad). Herkules marschiert zu einem der beiden Gegenstände..... bei Berührung erhebt er denselben, ab diesem Zeitpunkt, bestimmt die Stellung des Joysticks die Richtung des Wurfes und löst denselben auch aus. Wenn man sich zu lange zum "Zielen" aufhält, verschwindet der bereits zum Wurf erhobene Gegenstand ersatzlos. Ansonsten wäre ja das Spiel nicht zu einfach und man könnte die Ungeheuer leicht serienweise vernichten und Orakelgebühren sparen. Es ist komfortablerweise nicht notwendig, wenn man schräg nach oben in eine der beiden Richtungen werfen möchte, die genaue 45 Grad Stellung des Joysticks anzupeilen, nach rechts oder links gedrückt genügt es auch und bewirkt dasselbe. Oder für senkrecht nach oben, kann auch der Joystick nach unten gedrückt werden, der Schuß geht trotzdem nach oben los.
Dies ist eine wesentliche Erleichterung, da die TI-Joysticks keine Raststellungen aufweisen und die 45 Grad Stellung nicht immer leicht zu treffen ist, und dann der Wurf versehentlich in die falsche Richtung losgehen würde.

Wichtige Variablen:
SG: Schwierigkeitsgrad
RS: Restart
OO: Orakel übel gelaunt
G: Orakelgebühr
S: Geschwindigkeit
O$: Orakelspruch
D: Kontostand (Drachmen)
HD: Leitzahl für richtigen Tempel
OA: Orakelbestimmung:
Q: Orakeltest
A$, B$, C$: Segmente Sprite/Herkules

Sprites:
1: Herkules
2,3: Spinnen
4,5,6,7: Krokodile
8: Drachen
9: Zerberus
10: Feuer für Drachen
11,12: Waffen und Gegenstände

Ablauf:
110-170: Zeichendaten
200-220: Zeichen definieren
230-290: Bildschirmaufbau
300-370: Titel, Prolog, Abfrage SG
400-430: Diverse Variable
440-460: Tempel zeichnen
470-480: Sprites (Ungeheuer, Herkules)
500-520: Spielerschleife
540-590: Coinc Herkules/Ungeheuer (Auswertung)
610-680: Wurf nach Ungeheuer
690: Ungeheuer getroffen, Absturz
710: Coinc Herkules/Ungeheuer
720-800: Bewegung Herkules, Orakel fragen, Anzeige/Saldo
820: wie 710
830-850: Formänderung Ungeheuer, Drachen speit Feuer
870: wie 710
880: Formänderung Ungeheuer
900-920: Herkules geht schwimmen oder an Land
930-1220: Orakel
1230-1260: Spinne überfällt Herkules
1270-1300: Krokodil frißt Herkules auf
1310-1350: Drachen bespeit Herkules mit Feuer
1360-1400: Zerberus beißt Herkules
1410-1610: Tempelzerstörung, goldenes Vlies, Musik
1620-1650: Ende
1660-1700: SUB Herkules ist pleite

```
110 DATA 000000000040607,381
D0F070301,182C3E3E387CFDFD,F
FFEFCFCFCFC383E,0,000107
8F9FDF7E38,182C3E383E78FCFC,
FFFEFCFCFCFC383E
120 DATA 00E0B82E0B0255FF,0,
00003078CEFFFFFF,0,3C3C3C3C3
C3C3C7E,7E3C3C3C3C3C3C3C
,0000FFFF18FFFFFF,3F7FFFFF1E
FCF0C,FCFEFFFF783F0F03
130 DATA 48220A9004640112,0,
00000000FFAA55FF,0,00003078C
EFFFFFF,0,010701070F0107
1F,01071F3F013F7F01,B0E080E0
F080E0F8,B0E0F8FCB0FCFEB
140 DATA 031E071D01033BEB,3B
EF0F1F1F143622,808080001C979
CB7,F0E0FEFFFECA1B11,FFF
FFFFFFFFFFFF,0,000000E10C,0
0000631,031E030101033BEB
150 DATA 3B0F0F1F1F142448,B0
B0B0001C979CB,F2E1FEFEFED414
12,0,0003BCE01F001C03,0,
0EC00073806007,0103070703070
919,3F2B2A2928282828
160 DATA B0C0E0E0C0E09098,FC
D4549414141414,0103070703070
919,3F4B4A9190484824,80C
0E0E0C0E09098,FCD25289091212
24,030303193F4F4743
170 DATA 030303193F4F4743,B0
808030F8E8C884,030303193F4F4
743,80808030F8E8C884
180 !
190 CALL CLEAR :: CALL SCREE
N(8):: CALL MAGNIFY(3):: CAL
L COLOR(3,5,16,4,5,16,9,
13,8,10,15,5,11,13,4):: FOR
I=5 TO 8 :: CALL COLOR(I,16,
14):: NEXT I
200 RESTORE 110 :: FOR I=36
TO 47 :: READ A$ :: CALL CHA
R(I,A$):: NEXT I
210 FOR I=58 TO 64 :: READ A
$ :: CALL CHAR(I,A$):: NEXT
I :: FOR I=92 TO 124 ::
READ A$ :: CALL CHAR(I,A$)::
 NEXT I
220 FOR I=128 TO 134 STEP 2
:: READ A$ :: CALL CHAR(I,A$
):: NEXT I
230 FOR X=0 TO 1 :: FOR Z=0
TO 20 STEP 20 :: FOR J=0 TO
1 :: FOR I=2 TO 10 STEP
8 :: CALL HCHAR(2+X,I+J+Z,96
+(J*2)+X)
240 NEXT I :: NEXT J :: NEXT
 Z :: NEXT X :: ON WARNING N
EXT
250 FOR J=0 TO 1 STEP 1 :: F
OR I=12 TO 20 STEP 8 :: CALL
 HCHAR(3,I+J,96+(J*2)):: 
 NEXT I :: NEXT J
260 CALL HCHAR(4,1,104,192):
: CALL HCHAR(10,1,105,192)::
 CALL HCHAR(16,1,112,224
)
270 FOR J=23 TO 24 :: FOR I=
1 TO 31 STEP 30 :: CALL HCHA
R(J,I,112,2):: NEXT I ::
 NEXT J
280 CALL HCHAR(23,3,64,28)::
 CALL HCHAR(24,3,64,28)
290 FOR I=1 TO 40 :: FOR J=1
06 TO 107 :: CALL HCHAR(INT(
RND*6)+10,INT(RND*32)+1,
J):: NEXT J :: NEXT I
300 DISPLAY AT(5,6)SIZE(17):
"@H@E@R@K@U@L@E@S@" :: DISPL
AY AT(7,4)SIZE(21):"@SCH
WIERIGKEITSGRAD@@"
310 DISPLAY AT(17,5)SIZE(19)
:"@1@@@2000@DRACHMEN@" :: DI
SPLAY AT(19,5)SIZE(19):"
@2@@@1500@DRACHMEN@" :: DISP
LAY AT(21,5)SIZE(19):"@3@@@1
000@DRACHMEN@"
320 DISPLAY AT(24,1):"@@@@@@
@VON@GUSTAV@VLK@@@@@@@" :: A
CCEPT AT(22,14)BEEP SIZE
(1)VALIDATE("123"):SG :: DIS
PLAY AT(7,3)SIZE(22):"HOERE@
HERMES@BOTSCHAFT"
330 DISPLAY AT(17,1):"SCHREI
TE@UEBER@DEN@ACHERON@@O@HERK
ULES@UND@ZERSTOERE@DENTE
MPEL@DES@HADES@IN@WELCHEM@DA
S@GOLDENE@VLIES@VERBORGEN@"
340 DISPLAY AT(21,1):"DAS@WE
ISE@ORAKEL@SOLLST@DU@@BEFRAG
EN@UND@DEINE@FEINDE@@@@SI
ND@DEM@REICH@HADES@GEWISS@@@
@@@ACT@TASTE@DRUECKEN@@@@@"
350 CALL SOUND(500,110,0,440
,0,523,0):: FOR I=1 TO 300 :
: CALL KEY(1,K,ST):: IF
ST THEN 370
360 NEXT I
370 CALL HCHAR(17,1,112,192)
:: DISPLAY AT(23,1):RPT$("@"
,56):: IF ST=0 THEN 330
ELSE CALL HCHAR(5,1,104,96)
380 FOR I=17 TO 21 STEP 2 ::
 CALL HCHAR(I,7,112,19):: NE
XT I :: CALL HCHAR(22,16
,112)
390 !
400 A$="231206060404040C" ::
 B$="B480C0C04040406" :: C$=
"2312060C0C0B183" :: CAL
L CHAR(129,A$,131,B$,133,C$,
135,B$):: T1,T2,T3=1 :: RS=1
410 RANDOMIZE :: HD=INT(RND*
3)+1 :: IF SG=1 THEN D=2000
ELSE IF SG=2 THEN D=1500
 ELSE IF SG=3 THEN D=1000 ::
 X,Y=0
420 CALL CHAR(125,C$,126,"82
82821AFCE0C08",127,"B080C0C0
4040406")
430 R$="@DU@HAST@JETZT@NUR@M
EHR@@@@@@@@@@@@@@@@@@DRACHM
EN@@@@@@"
440 FOR J=1 TO 3 :: FOR I=0
TO 20 STEP 10 :: FOR Z=4 TO
8 STEP 4 :: CALL HCHAR(J
,Z+I,61-J,2):: NEXT Z :: NEX
T I :: NEXT J
450 FOR Z=6 TO 26 STEP 10 ::
 CALL HCHAR(1,Z,61):: CALL H
CHAR(1,Z+1,62):: NEXT Z
460 CALL SPRITE(#8,36,2,54,1
,0,12,#7,44,4,80,240,0,-12,#
6,92,4,80,120,0,-12,#5,4
4,15,104,180,0,-6)
470 CALL SPRITE(#4,92,15,104
,60,0,-6,#9,100,7,32,240,0,-
12,#2,116,2,124,1,0,8,#3
,120,16,124,128,0,8)
480 CALL SPRITE(#1,128,2,160
,128):: DISPLAY AT(23,1):"@@
@DAS@ORAKEL@VON@DELPHI@@
@@"
490 !
500 GOSUB 710
510 IF D<10 THEN CALL PL(RS)
520 GOSUB 820 :: GOSUB 710 :
: GOSUB 870 :: IF RS THEN 50
0 ELSE 400
530 !
540 ! COINC
550 CALL DISTANCE(#1,#11,DG)
:: CALL DISTANCE(#1,#12,DF):
: IF DG<324 OR DF<324 TH
EN RETURN
560 CALL MOTION(#2,0,0,#3,0,
0,#4,0,0,#5,0,0,#6,0,0,#7,0,
0,#8,0,0,#9,0,0,#10,0,0)
:: FOR J=2 TO 9
570 CALL COINC(#1,#J,20,C)::
 IF C THEN 590
580 NEXT J
590 IF J=2 OR J=3 THEN GOSUB
 1230 ELSE IF J=4 OR J=5 OR
J=6 OR J=7 THEN GOSUB 12
70 ELSE IF J=9 THEN GOSUB 13
60 ELSE GOSUB 1310
600 !
610 CALL COINC(ALL,C):: IF C
 THEN 620 ELSE RETURN
620 CALL COINC(#1,#12,16,C1)
:: CALL COINC(#1,#11,16,C2):
: IF C1 OR C2 THEN 630 E
LSE RETURN
630 CALL DELSPRITE(#10):: IF
 C1 THEN SP=12 ELSE SP=11
640 CALL MOTION(#1,0,0):: CA
LL POSITION(#1,YH,XH):: CALL
 LOCATE(#SP,YH-16,XH+8):
: CALL PATTERN(#1,124)
650 FOR M=1 TO 40 :: NEXT M
:: FOR Z=1 TO 20 :: CALL JOY
ST(1,X,Y):: IF X OR Y TH
EN 670
660 NEXT Z :: CALL DELSPRITE
(#SP):: RETURN
670 FOR T=1 TO 20 :: NEXT T
:: CALL MOTION(#SP,-18,X*4.5
):: FOR I=1 TO 4 :: FOR
J=2 TO 9 :: CALL COINC(#SP,#
J,20,C3):: IF C3 THEN 690
680 NEXT J :: NEXT I :: CALL
 DELSPRITE(#SP):: RETURN
690 CALL DELSPRITE(#SP):: FO
R I=1 TO 5 :: CALL MOTION(#J
,8,0):: CALL SOUND(100,1
000-(20*I),0):: NEXT I :: CA
LL DELSPRITE(#J):: RETURN
700 !
710 CALL COINC(ALL,C):: IF C
 THEN GOSUB 540
720 CALL JOYST(1,X,Y):: CALL
 MOTION(#1,-Y*2,X*2):: IF Y
OR X THEN CALL PATTERN(#
1,132)ELSE 790
730 CALL POSITION(#1,YH,XH):
: IF YH<8 THEN CALL LOCATE(#
1,8,XH)ELSE IF YH>160 TH
EN CALL LOCATE(#1,160,XH)ELS
E IF XH<8 THEN CALL LOCATE(#
1,YH,8)
740 IF XH>248 THEN CALL LOCA
```

```
TE(#1,YH,248)ELSE IF YH<116
AND YH>56 THEN GOSUB 900
750 IF YH>18 THEN 780
760 IF XH>40 AND XH<56 AND T
1 THEN XH=44 ELSE IF XH>120
AND XH<136 AND T2 THEN X
H=124 ELSE IF XH>200 AND XH<
216 AND T3 THEN XH=204 ELSE
780
770 GOSUB 1410
780 GOSUB 610 :: CALL PATTER
N(#1,128):: CALL MOTION(#1,0
,0):: CALL SOUND(5,-6,2)
:: IF XH>248 THEN CALL LOCAT
E(#1,YH,XH-6)
790 CALL KEY(1,K,ST):: IF ST
 THEN GOSUB 930 ELSE RETURN
800 D=D-G :: DISPLAY AT(23,1
):R$ :: DISPLAY AT(24,2)SIZE
(1+LEN(STR$(D))):D :: DI
SPLAY AT(24,2)SIZE(1):"@" ::
 CALL SOUND(300,110,0):: RET
URN
810 !
820 CALL COINC(ALL,C):: IF C
 THEN GOSUB 540
830 CALL PATTERN(#8,40,#7,44
,#6,92,#5,92,#4,44,#9,100,#2
,116,#3,120):: CALL POSI
TION(#8,YD,XD):: IF XD>158 T
HEN RETURN
840 IF XD=0 THEN RETURN
850 CALL SPRITE(#10,112,10,Y
D-8,XD+20,0,20):: CALL SOUND
(-300,-7,0):: RETURN
860 !
870 CALL COINC(ALL,C):: IF C
 THEN GOSUB 540
880 CALL PATTERN(#8,36,#7,92
,#6,44,#5,44,#4,92,#9,108,#2
,120,#3,116):: CALL DELS
PRITE(#10):: RETURN
890 !
900 CALL MOTION(#1,0,0):: IF
 YH<108 XOR YH>68 THEN 910 E
LSE 920
910 CALL CHAR(129,A$,131,B$,
133,C$,135,B$):: RETURN
920 CALL CHAR(129,"",131,"",
133,"",135,""):: RETURN
930 CALL DELSPRITE(#10):: OO
=0
940 D$="HOERE@MEINE@LEHREN@D
U@KANNST" :: E$="WAS@SOLL@IC
H@SAGEN@DU@DARFST" :: F$
="HERKULES@HUETE@DICH@ZEUS@@
@@" :: G$="GUT@ODER@SCHLECHT
@DEM@ZEUS@@"
950 RANDOMIZE :: OA=INT(RND*
8)+1 :: IF OA=Q THEN 950 ::
ON OA GOTO 960,970,980,9
90,1000,1010,1020,1030
960 O$=D$&"DICH@NUN@ZUM@HADE
S@SCHEREN@@" :: G=70 :: Q=1
:: OO=1 :: CALL DELSPRIT
E(#11,#12):: GOTO 1040
970 O$=D$&"NUN@GOLD@BEGEHREN
@@@@@@@@@@@" :: S=INT(RND*2)
+1 :: G=200 :: Q=2 :: GO
TO 1040
980 O$=E$&"MICH@JETZT@NUN@WE
ITER@FRAGEN" :: S=INT(RND*6)
+1 :: G=0 :: Q=3 :: GOTO
 1040
990 O$=E$&"ES@HEUTE@VIELLEIC
HT@WAGEN@@@" :: S=INT(RND*3)
+2 :: G=100 :: Q=4 :: GO
TO 1040
1000 O$=F$&"ZUERNT@DIR@ODER@
NICHT@@@@@@@" :: S=INT(RND*6
)+1 :: G=50 :: Q=5 :: GO
TO 1040
1010 O$=F$&"ZUERNT@DIR@FUERC
HTERLICH@@@@" :: S=INT(RND*2
)+5 :: G=200 :: Q=6 :: G
OTO 1040
1020 O$=G$&"IST@HEUTE@ALLES@
RECHT@@@@@@@" :: S=INT(RND*3
)+1 :: G=200 :: Q=7 :: G
OTO 1040
1030 O$=G$&"IST@HEUTE@GAR@NI
CHTS@RECHT@@" :: S=INT(RND*2
)+3 :: G=100 :: Q=8 :: G
OTO 1040
1040 DISPLAY AT(23,1):O$ ::
CALL SOUND(200,440,0,880,4,6
60,0):: CALL MOTION(#7,0
,-(4+(2*S)),#6,0,-(4+(2*S)),
#5,0,-(2+(INT(1.8*S))))
1050 CALL MOTION(#4,0,-(2+(I
NT(1.8*S))),#9,0,-(6+(INT(1.
5*S))),#2,0,8+INT(1.5*S)
,#3,0,8+INT(1.5*S))
1060 FOR N=1 TO 30 :: CALL J
OYST(1,X,Y):: IF X OR Y THEN
 1086
1070 NEXT N
1080 DISPLAY AT(23,1):"@DIE@
ORAKELGEBUEHR@BETRAEGT@@@@@@
@@@DRACHMEN@INCL@MWST@@"
 :: DISPLAY AT(24,2)SIZE(1+L
EN(STR$(G))):G :: DISPLAY AT
(24,2)SIZE(1):"@"
1090 IF OO=0 THEN 1110
1100 FOR T=1 TO 200 :: NEXT
T :: RETURN
1110 CALL DELSPRITE(#11,#12)
:: FOR I=1 TO 2 :: W=INT(RND
*5)+1 :: ON W GOTO 1120,
1130,1140,1150,1160
1120 W$(I)="1F0F07070F0F1F1F
1F0F0F0707030307F8F0E0E0F2F5
F9F9F9F5F2E0E0C0C0E" ::
P(I)=5 :: GOTO 1170
1130 W$(I)="000000073F"&RPT$
("EF",9)&"3F07000000E0FC"&RP
T$("F7",9)&"FCE" :: P(I)
=7 :: GOTO 1170
1140 W$(I)="010305"&RPT$("01
",13)&"00B040"&RPT$("0",26)
: P(I)=2 :: GOTO 1170
1150 W$(I)="010101070F070101
"&RPT$("01",8)&"82B79FFFFFFF
9F87B2"&RPT$("80",7):: P
(I)=5 :: GOTO 1170
1160 W$(I)="000103031A1D1D3D
3B7BEDEDDBDF6F3F000040F0FBF0
EC6E76B6F7EF9F7EBCF8" ::
 P(I)=15 :: GOTO 1170
1170 NEXT I
1180 CALL CHAR(136,W$(1),140
,W$(2))
1190 CALL SPRITE(#11,136,1,1
50,INT(RND*216)+16):: CALL D
ISTANCE(#11,#1,DI):: IF
DI<1200 THEN 1190
1200 CALL SPRITE(#12,140,1,1
50,INT(RND*216)+16):: CALL D
ISTANCE(#12,#11,E):: IF
E<1200 THEN 1200
1210 CALL DISTANCE(#12,#1,F)
:: IF F<1200 THEN 1200
1220 CALL COLOR(#11,P(1),#12
,P(2)):: RETURN
1230 RS=0 :: DISPLAY AT(23,1
):"@@EINE@GRAUSIGE@SPINNE@HA
T@@@@DICH@UEBERFALLEN@@@
@@@@@@@"
1240 CALL DELSPRITE(#10):: C
ALL POSITION(#1,Y,X):: FOR I
=1 TO 5 :: CALL PATTERN(
#J,120):: CALL LOCATE(#J,Y,X
):: CALL SOUND(-5,-2,0):: FO
R T=1 TO 40
1250 NEXT T :: CALL PATTERN(
#J,116):: CALL LOCATE(#J,Y-1
6,X):: FOR T=1 TO 40 ::
NEXT T :: NEXT I
1260 CALL DELSPRITE(ALL):: C
ALL E :: RETURN
1270 RS=0 :: DISPLAY AT(23,1
):"EIN@KROKODIL@VERSPEIST@DI
CH@JETZT@MIT@WOHLBEHAGEN
@@@@@@@"
1280 CALL POSITION(#1,Y,X)::
 CALL DELSPRITE(#1,#10):: FO
R I=1 TO 5 :: CALL PATTE
RN(#J,92):: CALL LOCATE(#J,Y
,X):: CALL SOUND(-5,-7,0)
1290 FOR T=1 TO 30 :: NEXT T
 :: CALL PATTERN(#J,44):: FO
R T=1 TO 30 :: NEXT T ::
 NEXT I
1300 CALL DELSPRITE(ALL):: C
ALL E :: RETURN
1310 RS=0 :: DISPLAY AT(23,1
):"@DER@BOESE@DRACHEN@SCHICK
T@@@DICH@JETZT@ZUM@HADES
@@@@@@@"
1320 CALL POSITION(#1,Y,X)::
 IF X<48 THEN X=48 ELSE IF X
>232 THEN X=232
1330 CALL DELSPRITE(#10):: C
ALL LOCATE(#1,Y,X,#8,Y,X-32)
:: FOR I=1 TO 5 :: CALL
PATTERN(#8,40):: CALL SPRITE
(#10,112,10,Y-8,X-22,0,20)
1340 CALL SOUND(-200,-7,0)::
 FOR T=1 TO 30 :: NEXT T ::
CALL PATTERN(#8,36):: CA
LL DELSPRITE(#10):: FOR T=1
TO 30 :: NEXT T :: NEXT I
1350 CALL DELSPRITE(ALL):: C
ALL E :: RETURN
1360 RS=0 :: DISPLAY AT(23,1
):"@DU@WIRST@JETZT@VOM@HOELL
EN@@@@HUND@ZERBERUS@GEBI
SSEN@@@"
1370 CALL DELSPRITE(#10):: C
ALL POSITION(#1,Y,X):: IF X<
32 THEN X=32 ELSE IF X>2
24 THEN X=224
1380 CALL LOCATE(#1,Y,X):: F
OR I=1 TO 5 :: CALL LOCATE(#
9,Y,X-16):: CALL SOUND(-
150,-2,0):: CALL PATTERN(#9,
100):: FOR T=1 TO 30 :: NEXT
 T
1390 CALL LOCATE(#9,Y,X+16):
: CALL SOUND(-150,-3,0):: CA
LL PATTERN(#9,108):: FOR
 T=1 TO 30 :: NEXT T :: NEXT
```

```
 I
1400 CALL DELSPRITE(ALL):: C
ALL E :: RETURN
1410 IF XH=44 THEN T1=0 ELSE
 IF XH=124 THEN T2=0 ELSE IF
 XH=204 THEN T3=0
1420 CALL DELSPRITE(#10):: C
ALL MOTION(#1,0,0):: CALL LO
CATE(#1,9,XH-2)
1430 X=(XH-12)/8 :: FOR I=X
TO X+5 :: CALL HCHAR(1,I,63)
:: CALL SOUND(-200,-7,0)
:: CALL PATTERN(#1,124):: FO
R T=1 TO 30
1440 NEXT T :: CALL PATTERN(
#1,128):: FOR T=1 TO 30 :: N
EXT T :: CALL HCHAR(1,I,
32):: NEXT I
1450 FOR J=2 TO 3 :: FOR I=0
 TO 1 :: CALL HCHAR(J,X+I,63
):: CALL PATTERN(#1,128)
:: CALL SOUND(-200,-6,0):: F
OR T=1 TO 30 :: NEXT T
1460 CALL PATTERN(#1,132)::
CALL HCHAR(J,X+I,32):: FOR T
=1 TO 30 :: NEXT T :: NE
XT I :: NEXT J
1470 FOR J=2 TO 3 :: FOR I=4
 TO 5 :: CALL HCHAR(J,X+I,63
):: CALL PATTERN(#1,128)
:: CALL SOUND(-200,-5,0):: F
OR T=1 TO 30 :: NEXT T
1480 CALL PATTERN(#1,124)::
CALL HCHAR(J,X+I,32):: FOR T
=1 TO 30 :: NEXT T :: NE
XT I :: NEXT J
1490 IF HD=1 AND T1=0 THEN 1
500 ELSE IF HD=2 AND T2=0 TH
EN 1500 ELSE IF HD=3 AND
T3=0 THEN 1500 ELSE 1510
1500 DISPLAY AT(23,1):"DER@T
EMPEL@DES@HADES@IST@HINDAS@G
OLDENE@VLIES@IST@DEIN@@"
 :: RS=0 :: GOTO 1530
1510 CALL SOUND(1000,220,0,3
30,0,523,0):: D=D-500 :: IF
D<1 THEN RETURN
1520 DISPLAY AT(23,1):R$ ::
DISPLAY AT(24,2)SIZE(1+LEN(S
TR$(D))):D :: DISPLAY AT
(24,2)SIZE(1):"@" :: CALL LO
CATE(#1,160,120):: RETURN
1530 FOR I=2 TO 12 :: CALL D
ELSPRITE(#I):: NEXT I :: CAL
L CHAR(136,"00180808 0C0C
6FFFFF6F0C0C0808180000 0C0404
0C1CFEFFFFFE1C0C04040C")
1540 CALL POSITION(#1,Y,X)::
 CALL SPRITE(#11,136,11,Y+20
,X)
1550 DATA 6,466,2,349,8,349,
8,466,6,392,2,392,16,392,6,4
40,2,466,8,523,8,440,4,4
66,4,587,16,466
1560 DATA 6,440,2,466,8,523,
8,523,4,587,4,466,8,523,8,52
3,6,440,2,466,8,523,8,52
3,4,587,4,466,8,523
1570 DATA 8,523,6,466,2,440,
4,392,4,622,4,587,4,523,8,46
6,8,523,8,587
1580 DATA 6,466,2,440,4,392,
4,622,4,587,4,523,8,587,8,52
3,8,466
1590 FOR J=1 TO 2 :: RESTORE
 1550 :: FOR I=1 TO 14 :: RE
AD A,B :: GOSUB 1610 ::
NEXT I :: NEXT J :: FOR I=1
TO 34 :: READ A,B :: GOSUB 1
610
1600 NEXT I :: CALL E :: RET
URN
1610 CALL SOUND(A*50,B,0,B-2
,10,B+2,10):: RETURN
1620 SUB E
1630 DISPLAY AT(23,1):"@@NEU
ES@SPIEL@START@MIT@ACT@@@TAS
TE@UND@BITTE@WARTEN@@@@"
1640 CALL KEY(1,K,ST):: IF S
T THEN 1650 ELSE 1640
1650 CALL DELSPRITE(ALL):: D
ISPLAY AT(23,1):RPT$("@",56)
:: FOR D=1 TO 300 :: NEX
T D :: SUBEND
1660 SUB PL(RS):: RS=0
1670 CALL DELSPRITE(ALL):: D
ISPLAY AT(23,1):"@ZEUS@VERWA
NDELT@DICH@ZUR@@@@STRAFE
@IN@EINEN@PLEITEGEIER"
1680 A$="030509112828280 0" :
: B$="C0A0908814141 4" :: CAL
L CHAR(136,"01010111397D
EFAB"&A$&"B060A00B9CBEF7D5"&
B$)
1690 CALL CHAR(140,"0141E1F1
B9BD2B0B"&A$&"B062A70F9DADD4
D0"&B$):: CALL SPRITE(#1
2,136,2,40,120):: FOR I=1 TO
 10 :: CALL PATTERN(#12,140)
1700 CALL SOUND(-5,-1,0):: F
OR D=1 TO 50 :: NEXT D :: CA
LL PATTERN(#12,136):: FO
R D=1 TO 50 :: NEXT D :: NEX
T I :: CALL E :: SUBEND
```

Computer an Schulen – Verschlafen wir den Anschluß?

Wenn Sie auch CPU-Leser sind, so werden Sie sich wohl an unseren Schulbericht vom September erinnern, in dem wir zwei hessische Gymnasien, die Herderschule in Gießen und das Oberstufengymnasium in Kassel/Oberzwehren, mit ihren Computererfahrungen vorgestellt hatten.
Die Ergebnisse von damals erhärteten sich nun in einer groß angelegten Fragebogen-Aktion, die HC gemeinsam mit der Zeitschrift CHIP und den bekannten Büromaschinen-Hersteller Triumph-Adler durchführte. Sage und schreibe 20.000 Schulen wurden angeschrieben.

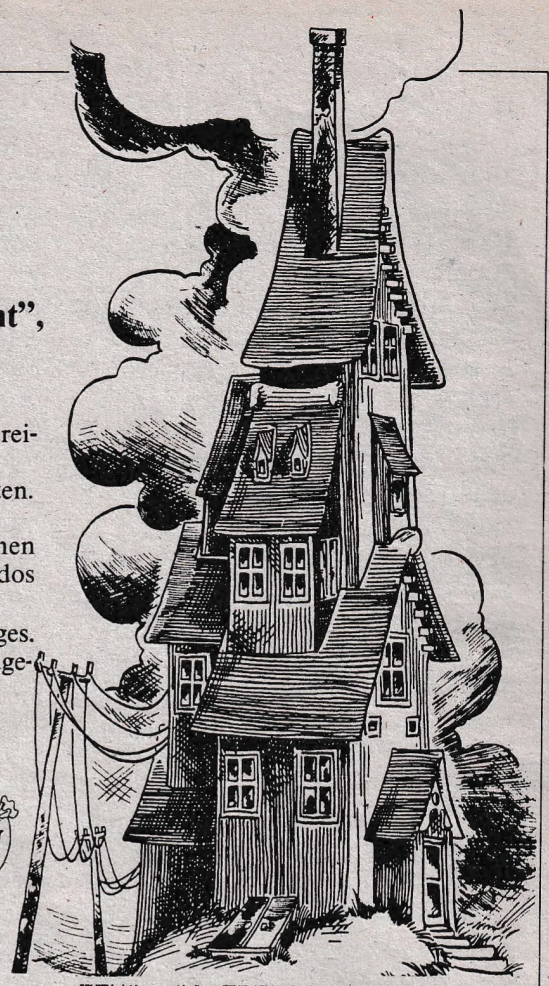
3.357 Schulen haben den diferenzierten Fragebogen beantwortet und nahmen dadurch automatisch an der Verlosung von 200 alphatronic-Mikrocomputern der Triumph-Adler AG teil. Die Ergebnisse der Aktion wurden jetzt der Bundesministerin für Bildung und Wissenschaft, Fr. Dorothee Wilms, in Bonn übergeben.
Es sieht im internationalen Vergleich nicht gut für den deutschen Nachwuchs am Computer aus. "Wir haben", so Richard Kerler, Redaktionsdirektor im Vogel-Verlag, Würzburg, "fast ein Jahrzehnt verschlafen". Wörtlich sagte Kerler: "Wir haben in unserem Lande die verlorene Generation der 30 bis 40jährigen. Und auch die junge Generation wird nicht systematisch an den Computer herangeführt. Das beweisen auch die Ergebnisse. Nur 60,5 Prozent der Schulen, die an der Umfrage teilgenommen haben, benutzen bereits einen Computer. Ganz vorne stehen die Gymnasien mit 90 Prozent. Das Schlußlicht bilden die Hauptschulen mit 25 Prozent. Also wieder das übliche Bildungsgefälle. Bei der Aufteilung nach Bundesländern steht Berlin mit 89,1 Prozent vor Rheinland-Pfalz mit 68,6 Prozent und Bayern mit 65,4 Prozent vorne. Schlußlicht ist Niedersachsen mit 52,5 Prozent. Beängstigend sieht es bei den Lehrkräften aus. Nur 25,6 Prozent der Schulen, die mit Mikrocomputern arbeiten, verfügen über sechs oder mehr Lehrkräfte, die mit den Geräten vertraut sind. Aber: In fast 50 Prozent der Schulen stehen nur ein bis drei Lehrkräfte zur Verfügung. Und das bei oft 2.000 Schülern und mehr.

Horror House
für den TI-99/4A

Horror House – Ein Abenteuerspiel wie es "im Buche steht", bei dem mehrere Wege zum Ziel führen.

Als glücklicher Erbe einer alten Villa, mit sage und schreibe 48 Zimmern, treten Sie voller Neugier die erste Hausbesichtigung an, bis leider – oh Schreck – die schwere, alte Haustür ins Schloß fällt und sich auch nicht durch jegliche Anstrengung mehr öffnen läßt. Ihre Situation ist eindeutig – Sie sitzen in der Falle.
Doch ganz so aussichtslos, wie in der ersten Schrecksekunde angenommen, ist die Lage doch nicht. Es befindet sich nämlich irgendwo im Haus ein Schlüssel und außerdem noch ein Schatz – der Geldkelch Ihrer Urahnen.
Finden Sie beides, können Sie das Haus wieder verlassen.
Mehr wollen wir aber jetzt nicht verraten – sehen Sie selbst, wie es weitergeht!

Spielanleitung:
100-460: Spielregeln und Vorbereitung
470-1040: Einlesen der Hausdaten. Verteilen der Gegenstände
1050-1360: Ausgabe des momentanen Raumes und Eingabe der Kommandos
1370-1430: Nase aufgeschlagen
1440-1650: Überprüfen des Zuges. welche Gegenstände sind dazugekommen
1660-1760: Ausrüstung auflisten
1770-1960: Todesarten
1970-2040: Todesmusik
2050-2200: Treppengehen
2350-2410: Spiel gelöst
2420-2620: Ende



```
60 REM  Horror House
        Von
70 REM  Norbert Schoenfeld

80 REM

90 REM
100 CALL CLEAR
110 CALL CHAR(64,"007E9292FE
9292FC")
120 PRINT TAB(8);"HORROR HOU
SE":TAB(7);"==============":
::TAB(12);"Von"::TAB(4);
"Norbert Schoenfeld @":::::
130 PRINT "WOLLEN SIE DIE RE
GELN ? J/N":::
140 CALL KEY(3,K,S)
150 IF K=78 THEN 400
160 IF K<>74 THEN 140
170 PRINT "SIE SIND IN EINEM
 ALTEN HAUSUND MUESSEN VERSU
CHEN HERAUSZU KOMMEN.":"
DAZU MUESSEN SIE DEN HAUS-"
180 PRINT "TUERSCHLUESSEL UN
D EINEN    GOLDKELCH FINDEN,
DIE IRGEND-WO IM HAUS VE
RSTECKT SIND."
190 PRINT "SIE KOMMEN ABER N
UR DURCH   OFFENE TUEREN.":"
DURCH GESCHLOSSENE TUERE
N   KOMMEN SIE MIT DEM DIETR
ICH,"
200 PRINT "DURCH VERKLEMMTE
MIT DEM    BRECHEISEN,":"UND
 DURCH EINGEROSTETE MIT
 HAMMER UND OELKANNE.        >
>"::
210 CALL KEY(3,K,S)
220 IF S=0 THEN 210
230 PRINT ::::"LOECHER IM BO
DEN KOENNEN SIEMIT DER TASCH
ENLAMPE-DIE NA-TUERLICH
BATTERIEN BRAUCHT-"
240 PRINT "SEHEN UND DEN VAM
PIR TOETEN SIE MIT EINEM KRE
UZ.":"VERGESSEN SIE AUCH
 NICHT"
250 PRINT "FUER LEBENSMITTEL
 MIT DOSEN-OEFFNER ZU SORGEN
,DA SIE    SONST NACH 60
 SCHRITTEN VER-HUNGERN.":::
260 PRINT "SOBALD SIE DEN GO
LDKELCH UNDDEN HAUSSCHLUESSE
L HABEN    KOENNEN SIE D
URCH DIE HAUS-"
270 PRINT "TUER DAS HAUS VER
LASSEN.":::"TREPPEN GIBT ES I
M FLUR 1,  FLUR 2 UND FL
UR 3,DIE UEBER-"
280 PRINT "EINANDER LIEGEN.
          >>":::
290 CALL KEY(3,K,S)
300 IF S=0 THEN 290
310 PRINT :::::"DIE KOMMANDO
S:":::"N=SCHRITT NACH NORDEN"
::"S=SCHRITT NACH SUEDEN
":::"W=SCHRITT NACH WESTEN"::
320 PRINT "O=SCHRITT NACH OS
TEN"::"A=TREPPE RUNTER"::"R=
TREPPE RAUF"::"T=AUSRUES
TUNG AUFLISTEN"::TAB(26);">>
"::
330 CALL KEY(3,K,S)
340 IF S=0 THEN 330
350 PRINT ::::"AM ANFANG WER
DEN SIE AUFGE- FORDERT,EINE
RAUMVERTTEI-  LUNGSZAHL
 EINZUGEBEN."
360 PRINT "DIESE DIENT DAZU
DIE GEGEN- STAENDE IM HAUS Z
U VER-     TEILEN."
370 PRINT "WENN SIE DIE SELB
E ZAHL EIN-GEBEN,WIRD DIE VE
RTEILUNG   NICHT VERAEND
ERT.":::::TAB(26);">>"
380 CALL KEY(3,K,S)
390 IF S=0 THEN 380
400 CALL SCREEN(2)
410 FOR I=1 TO 16
420 CALL COLOR(I,15,1)
430 NEXT I
440 CALL CLEAR
450 OPTION BASE 1
460 DIM NR1(48),NR2(48),NA$(
48),BE$(48),IN$(14),WA$(9),N
R3(12),COD(14),RN(13)
470 DATA 100000203,30001188,
FOLTERKAMMER,101000300,30008
181,KERKER 3,102010406,2
0008883,KERKER 2,103000500
480 DATA 10008181,KERKER 1,1
04000007,10008115,KOHLENKELL
```

```
ER,100030009,62001316,KA
RTOFFELKELLER,100050800
490 DATA 81001551,FLUR 1,107
000010,10005115,HOBBYKELLER,
100061000,40001641,WEINK
ELLER,109080000,10004511
500 DATA VORRATSKELLER,10000
1215,20001122,KLAVIERZIMMER,
111000016,30002113,WOHNZ
IMMER,112001417,70006136
510 DATA FLEISCHKAMMER,11300
0017,40003117,SPEISEKAMMER,1
00111620,10001245,BIBLIO
THEK,115121721,40004336
520 DATA SPEISESAAL,11614182
1,40003744,KUECHE,117001922,
80144167,SPUELKUECHE,118
000023,50006116,TROCKENRAUM
530 DATA 100152125,83009556,
VORRAUM,120172226,81005452,F
LUR 2,121182329,10005742
,AUFENTHALTSRAUM,122190048
540 DATA 80004615,WASCHKUECH
E,100002500,40001141,WINTERG
ARTEN,124202600,10004621
,KONZERTSAAL,125212800,10002
271
550 DATA SCHULZIMMER,1002129
28,70001633,BAD(PERSONAL),12
6270000,40007311,WEIBL.P
ERSONAL,127224800,10003271
560 DATA MAENNL.PERSONAL,100
003135,40001144,EDGAR,130003
236,20004125,SPIELZIMMER
,131003437,30002147,WILLIAM
570 DATA 142000034,50006116,
ARBEITSZIMMER,132333338,4000
4667,AMME,100303643,4000
1446,GRAEFIN,135313744,10004
525
580 DATA FLUR 4,136323840,10
002733,DIANA,137343941,62003
734,BLUMENZIMMER,1383300
42,72003613,GUMMIZELLE,13637
4146
590 DATA 81007322,FLUR 3,140
384200,82002431,WARTEZIMMER,
141393300,50003361,SEKRE
TARIAT,100354447,50001664,GR
AF
600 DATA 143364545,10006524,
ANKLEIDERAUM,144444600,10002
451,BUTLER,145400000,100
05211,SEKRETAERIN,100434500
610 DATA 62001461,BAD/GRAF,1
29230000,62007511,HEIZUNGSRA
UM
620 DATA WAND,HOLZTUER,HOLZT
UER VERKLEMMT,HOLZTUER GESCH
LOSSEN,EISENTUER,EISENTU
ER EINGEROSTET
630 RESTORE 440
640 PRINT "BITTE WARTEN SIE"
::::::::
650 FOR I=1 TO 48
660 READ NR1(I),NR2(I),NA$(I
)
670 NEXT I
680 FOR I=1 TO 9
690 READ WA$(I)
700 NEXT I
710 FOR I=1 TO 14
720 READ IN$(I)
730 NEXT I
740 FOR I=1 TO 48
750 READ BE$(I)
760 NEXT I
770 FOR I=1 TO 14
780 READ COD(I)
790 NEXT I
800 IF VZ=0 THEN 860
810 FOR I=1 TO 13
820 IF I=9 THEN 840
830 NR2(RN(I))=NR2(RN(I))+10
000*I
840 NEXT I
850 GOTO 970
860 PRINT "GEBEN SIE JETZT I
HRE RAUM-"::"VERTEILUNGSZAHL
 EIN !"::
870 INPUT A
880 RANDOMIZE A
890 FOR I=1 TO 14
900 IF COD(I)=8 THEN 960
910 B=INT(RND*48)+1
920 IF SEG$(STR$(NR2(B)),3,2
)<>"00" THEN 910
930 IF COD(I)<>VAL(SEG$(STR$
(NR2(B)),1,1))THEN 910
940 NR2(B)=NR2(B)+10000*I
950 RN(I)=B
960 NEXT I
970 FOR I=1 TO 12
980 NR3(I)=0
990 NEXT I
1000 VZ=0
1010 SC=0
1020 RA=20
1030 IN=0
1040 A$=STR$(NR1(RA))
1050 B$=STR$(NR2(RA))
1060 CALL CLEAR
1070 PRINT NA$(RA):::
1080 PRINT "WEST:";WA$(VAL(S
EG$(B$,5,1)))::"SUED:";WA$(V
AL(SEG$(B$,6,1)))::
1090 PRINT "OST :";WA$(VAL(S
EG$(B$,7,1)))::"NORD:";WA$(V
AL(SEG$(B$,8,1)))):::
1100 PRINT "BEMERKUNGEN:":BE
$(RA)::
1110 IF IN=0 THEN 1130
1120 PRINT "INHALT:";IN$(IN)
::IN$(IN);" WURDE ";"AUFGEND
MMEN."
1130 PRINT ::
1140 CALL VCHAR(1,31,64,24)
1150 CALL KEY(3,K,S)
1160 CALL SOUND(1,1100,20)
1170 IF K=84 THEN 1660
1180 IF (K=65)+(K=82)THEN 20
50
1190 IF (K<>78)*(K<>83)*(K<>
87)*(K<>79)THEN 1150
1200 SC=SC+1
1210 IF K<>87 THEN 1230
1220 K=5
1230 IF K<>83 THEN 1250
1240 K=6
1250 IF K<>79 THEN 1270
1260 K=7
1270 IF K<>78 THEN 1290
1280 K=8
1290 A=VAL(SEG$(B$,K,1))
1300 B=VAL(SEG$(A$,(K-4)*2,2
))
1310 IF A=1 THEN 1370
1320 IF (A=2)+(A=5)+(A=8)THE
N 1440
1330 IF ((A=4)+(A=7))*(NR3(3
)=1)THEN 1440
1340 IF (A=6)*(NR3(4)=1)*(NR
3(5)=1)THEN 1440
1350 IF (A=3)*(NR3(2)=1)THEN
 1440
1360 IF (A=9)*(NR3(10)=1)*(N
R3(11)=1)THEN 2350
1370 FOR I=1 TO 30
1380 CALL SOUND(-50,110,I,11
4,I,118,I,-5,I)
1390 NEXT I
1400 PRINT ::"SIE HABEN SICH
 DIE NASE"::"AUFGESCHLAGEN !
"::"SEIEN SIE VORSICHTIG
ER !!"
1410 GOSUB 2580
1420 IF (SC=60)*((NR3(6)<>1)
+(NR3(7)<>1))THEN 1940
1430 GOTO 1040
1440 Z=0
1450 RA=B
1460 A$=STR$(NR1(RA))
1470 B$=STR$(NR2(RA))
1480 B=VAL(SEG$(B$,3,2))
1490 A=VAL(SEG$(B$,2,1))
1500 IF (A=2)*((NR3(8)<>1)+(
NR3(12)<>1))THEN 1770
1510 IF (B=14)*(NR3(9)=1)THE
N 1870
1520 IF (B=13)*(NR3(1)<>1)TH
EN 1910
1530 IF (SC=60)*((NR3(6)<>1)
+(NR3(7)<>1))THEN 1940
```

```
1540 IN=0
1550 IF (B=0)+(B=14)THEN 165
0
1560 IN=B
1570 IF B<>13 THEN 1630
1580 NR2(RA)=NR2(RA)-13*1000
0
1590 IN=9
1600 PRINT "IM RAUM IST EIN
VAMPIR.":"SIE TOETEN IHN MI
T DEM KREUZ":"UND NEHME
N EINEN ASCHENHAU-":"FEN MI
T.":
1610 GOSUB 2580
1620 GOTO 1640
1630 NR2(RA)=NR2(RA)-IN*1000
0
1640 NR3(IN)=1
1650 GOTO 1040
1660 CALL CLEAR
1670 PRINT "AUSRUESTUNG:
        ";STR$(SC):"--------
----":::
1680 FOR I=1 TO 12
1690 IF NR3(I)=0 THEN 1710
1700 PRINT IN$(I),
1710 NEXT I
1720 PRINT :::::
1730 CALL KEY(3,K,S)
1740 CALL SOUND(1,2000,20)
1750 IF S=0 THEN 1730
1760 GOTO 1040
1770 PRINT "SIE SIND IN EIN
LOCH GE-":"FALLEN UND HABEN
 SICH"::
1780 IN=0
1790 IF RND>0.38 THEN 1840
1800 PRINT "IHRE NASE GEBROC
HEN.":::
1810 GOSUB 2580
1820 RA=22
1830 GOTO 1040
1840 PRINT "IHR GENICK GEBRO
CHEN.":::
1850 GOSUB 2580
1860 GOTO 1970
1870 PRINT ::::"SIE HABEN SI
CH AN EINEM"::"MESSER VERLET
ZT,WODURCH DER"::"VAMPIR
 WIEDER AUFGEWACHT IST"::
1880 PRINT "UND SIE DANN GET
OETET HAT.":::
1890 GOSUB 2580
1900 GOTO 1970
1910 PRINT ::::"EIN VAMPIR H
AT SIE GEBISSEN,"::"SODASS S
IE STERBEN.":::
1920 GOSUB 2580
1930 GOTO 1970
1940 PRINT ::::"LEIDER SIND
SIE VERHUNGERT,"::"DA SIE KE
INE LEBENSMITTEL"::"MEHR
 HABEN.":::
1950 GOSUB 2580
1960 GOTO 1970
1970 CALL CLEAR
1980 FOR I=120 TO 0 STEP -1
1990 CALL SOUND(-100,110+I,I
NT(I/8))
2000 NEXT I
2010 PRINT "SIE SIND GESTORB
EN !":::::
2020 GOSUB 2580
2030 PRINT "FRIEDE IHRER ASC
HE.":::::
2040 GOTO 2430
2050 IF (RA=21)+(RA=7)+(RA=4
0)>-1 THEN 1040
2060 IF K=65 THEN 2140
2070 IF RA=7 THEN 2100
2080 IF RA=21 THEN 2120
2090 GOTO 1040
2100 RA=21
2110 GOTO 1040
2120 RA=40
2130 GOTO 1040
2140 IF RA=40 THEN 2170
2150 IF RA=21 THEN 2190
2160 GOTO 1040
2170 RA=21
2180 GOTO 1040
2190 RA=7
2200 GOTO 1040
2210 DATA EISENTUER GESCHLOS
SEN,GITTERTUER,HAUSTUER,KREU
Z,BRECHEISEN,DIETRICH,OE
LKANNE,HAMMER,LEBENSMITTEL
2220 DATA DOSENOEFFNER,TASCH
ENLAMPE,ASCHENHAUFEN,HAUSSCH
LUESSEL,GOLDKELCH,BATTER
IEN,VAMPIR,MESSER
2230 DATA EIN BEIL HAENGT AN
 DER WAND,KNOCHEN LIEGEN AM
BODEN,KETTEN RASSELN ???
,SKELETTE LIEGEN IN DER ECKE
2240 DATA EIN FEUER GLUEHT,E
INE MAUS LAEUFT UMHER,NICHTS
,DER BODEN KLEBT,SCHLANG
EN KRIECHEN HERUM
2250 DATA WASSER TROPFT VON
DER DECKE,DAS KLAVIER SPIELT
 ?!?,AUF DEM TISCH STEHT
 EIN GLAS,ES STINKT STARK !
2260 DATA EINE MAUS LIEGT AM
 BODEN,NICHTS,ES IST ANGERIC
HTET ?!?,DER WASSERHAHN
TROPFT,SIE SEHEN EIN MESSER
2270 DATA ES ZIEHT !!!
2280 DATA DER RAUM IST ERLEU
CHTET,EINE TUER KLAPPERT,IN
DER DECKE IST EIN LOCH,S
IE SEHEN EIN GRAB !!!,NICHTS
2290 DATA ALLE BILDER WACKEL
N,MAN HOERT EIN STOEHNEN,DAS
 WASSER LAEUFT,DER RAUM
IST VERSPIEGELT
2300 DATA FLASCHEN LIEGEN AM
 BODEN,NICHTS,EIN TEDDY LIEG
T AM BODEN,DER FENSTERLA
DEN KLAPPERT,IM RAUM IST...
2310 DATA MAN HOERT EIN SEUF
ZEN,GIFT LIEGT AUF DEM STUHL
,NICHTS,EIN BABY WEINT ?
!?,DIE SONNE SCHEINT
2320 DATA EIN LOCH IST IM BO
DEN,NICHTS,EIN LOCH IST IM B
ODEN,DER TISCH IST VERSC
HLOSSEN,DAS BETT IST LEER
2330 DATA NICHTS,IM SCHRANK
IST EIN GEWEHR !,DAS BETT IS
T BLUTROT,EIN RING LIEGT
 AM BODEN,ES STINKT NACH OEL
2340 DATA 1,6,3,4,4,1,4,5,8,
7,6,5,2,8
2350 CALL CLEAR
2360 CALL SOUND(100,262,0,33
0,0,392,0)
2370 CALL SOUND(100,330,0,39
2,0,523,0)
2380 CALL SOUND(100,392,0,52
3,0,659,0)
2390 FOR I=0 TO 30
2400 CALL SOUND(-50,523,0,65
9,I,784,I)
2410 NEXT I
2420 PRINT "SIE HABEN DAS SP
IEL GELOEST,"::"UND DAZU";SC
;"SCHRITTE"::"BENOETIGT.
":::
2430 PRINT "VERSUCHEN SIE DO
CH IHR"::"ERGEBNIS ZU VERBES
SERN."::"NOCHMAL:=<J>"::
"ENDE    :=<N>"::::
2440 CALL KEY(3,K,S)
2450 CALL SOUND(1,1100,20)
2460 IF K=78 THEN 2610
2470 IF K<>74 THEN 2440
2480 PRINT "WOLLEN SIE"::"<A
> EINE NEUE RAUMVERTEIL-"::"
    UNGSZAHL EINGEBEN OD
ER"::
2490 PRINT "<B> DIE VERTEILU
NG SO"::"    BELASSEN ?"::::
2500 CALL KEY(3,K,S)
2510 CALL SOUND(1,1100,20)
2520 IF K=65 THEN 630
2530 IF K<>66 THEN 2500
2540 VZ=1
2550 GOTO 630
2560 CALL CLEAR
2570 END
2580 FOR I=1 TO 600
2590 NEXT I
2600 RETURN
2610 CALL CLEAR
2620 END
```

Pilot
für den TI-99/4A

Diese hervorragende Version eines Landsimulators nutzt alle technischen und softwaremäßig eingebauten Fähigkeiten Ihres TI mit EX-Basic. Alle notwendigen Anleitungen gibt das Spiel selbst aus.

```
100 REM PROGRAMMIERT VON
          HORST FRANKE
210 REM
220 GOSUB 1000
230 GOTO 2000
240 GOSUB 3000
260 GOSUB 4000
270 GOTO 220
300 REM
1000 REM INITIALISIERUNG
1010 REM
1020 REM
1030 CALL DEFINITION
1040 CALL CLEAR
1050 CALL SCREEN(2)
1060 CALL SPRITE(#1,123,14,2
4,40)
1070 CALL SPRITE(#2,124,11,2
4,32)
1075 CALL SPRITE(#3,124,11,2
4,24)
1080 CALL SPRITE(#4,125,14,2
4,16)
1085 LET C=3
1090 CALL MOTION(#1,0,C,#2,0
,C,#3,0,C,#4,0,C)
1100 REM
1120 LET SP=1
1135 CALL COLOR(2,14,11)
1140 DISPLAY AT(08,6):"*** *
 *   *** ***"
1150 DISPLAY AT(09,6):"* * *
 *   * *  * "
1160 DISPLAY AT(10,6):"*** *
 *   * *  * "
1170 DISPLAY AT(11,6):"*   *
 *   * *  * "
1180 DISPLAY AT(12,6):"*   *
 *   * *  * "
1190 DISPLAY AT(13,6):"*   *
 *** ***  * "
1192 FOR I=1 TO 500
1193 CALL SOUND(5,-5,0)
1194 NEXT I
1195 FOR I=1 TO 8
1196 CALL COLOR(I,16,1)
1197 NEXT I
```

```
1198 CALL COLOR(13,7,2)
1200 DISPLAY AT(15,3):"WUENS
CHEN SIE EIN SPIELAN-"
1210 DISPLAY AT(17,9):"LEITU
NG ? (J/N)"
1230 ACCEPT AT(17,25)BEEP VA
LIDATE("J","N")SIZE(1):A$
1240 IF A$="J" THEN CALL SPI
ELANLEITUNG
1999
2000 REM VORPROGRAMM
2010 REM
2020 REM
2025 CALL CLEAR
2030 DISPLAY AT(10,4):"WIEVI
ELE MITSPIELER"
2040 DISPLAY AT(11,4):"TASTE
 DRUECKEN (1-4)"
2050 ACCEPT AT(11,24)BEEP VA
LIDATE("1","2","3","4"):A
2065 CALL CLEAR
2070 DISPLAY AT(10,4):"FUER
START"
2080 PRINT "ENTERTASTE DRUEC
KEN >>>>>>"
2090 CALL KEY(3,KEY,STATUS)
2100 IF STATUS=0 THEN 2090
2110 IF KEY<>13 THEN GOTO 20
90
2120 CALL CLEAR
2125 LET S=1
2126 LET T=35
2130 CALL HCHAR(24,1,129,32)
2140 CALL SPRITE(#1,123,14,2
4,40)
2150 CALL SPRITE(#2,124,11,2
4,32)
2160 CALL SPRITE(#3,124,11,2
4,24)
2170 CALL SPRITE(#4,125,14,2
4,16)
2190 CALL MOTION(#1,0,C,#2,0
,C,#3,0,C,#4,0,C)
2200 CALL HCHAR(2,1,61,32)
2999 GOTO 240
3000 REM HAUPTPROGRAMM
3010 REM
```

```
3015 DISPLAY AT(1,3):"SPIELE
R:";S;" TREIBSTOFF:";T
3020 CALL SPRITE(#6,127,8,16
,100)
3030 FOR G=0 TO 22
3040 IF G<-20 THEN LET G=-20
3050 IF G>19 THEN LET G=19
3060 CALL DISTANCE(#6,240,10
0,D)
3070 IF D<40 THEN LET G=1
3080 CALL MOTION(#6,G,0)
3090 CALL DISTANCE(#6,#1,D)
3100 IF D<25 THEN GOSUB 3210
3110 CALL DISTANCE(#6,180,10
0,D)
3120 IF D<25 THEN GOSUB 3300
3130 IF T=0 THEN 3200
3140 CALL KEY(1,KEY,STATUS)
3150 IF STATUS=0 THEN 3200
3160 LET G=G-1
3170 LET T=T-1
3180 DISPLAY AT(1,26):T
3190 GOTO 3040
3200 NEXT G
3210 REM ENERGIE AUFNEHMEN
3220 CALL MOTION(#6,0,0,#1,0
,0,#2,0,0,#3,0,0,#4,0,0)
3230 FOR I=150 TO 1000 STEP
50
3240 CALL SOUND(20,I,1,I*1.5
,2,I*2.25,3)
3250 NEXT I
3260 LET T=35
3270 DISPLAY AT(1,26):T
3280 CALL MOTION(#1,0,C,#2,0
,C,#3,0,C,#4,0,C)
3290 RETURN
3300 IF G>3 THEN 3450
3310 CALL MOTION(#6,0,0)
3320 FOR I=500 TO 1500 STEP
500
3330 CALL SOUND(100,I,0)
3340 NEXT I
3350 LET C=C+.5
3360 CALL MOTION(#1,0,C,#2,0
,C,#3,0,C,#4,0,C)
3370 IF C>10 THEN C=10
3380 LET G=-15
3390 CALL KEY(1,KEY,STATUS)
3400 IF STATUS=0 THEN 3390
3410 CALL MOTION(#6,-15,0)
3420 FOR I=1 TO 100
3430 NEXT I
3440 RETURN
3450 REM ABSTURZ
3460 CALL DELSPRITE(#6)
3470 CALL SPRITE(#7,128,16,1
75,100)
3480 CALL SOUND(1500,-5,0)
3490 FOR I=1 TO 10
3500 CALL PATTERN(#7,128)
```

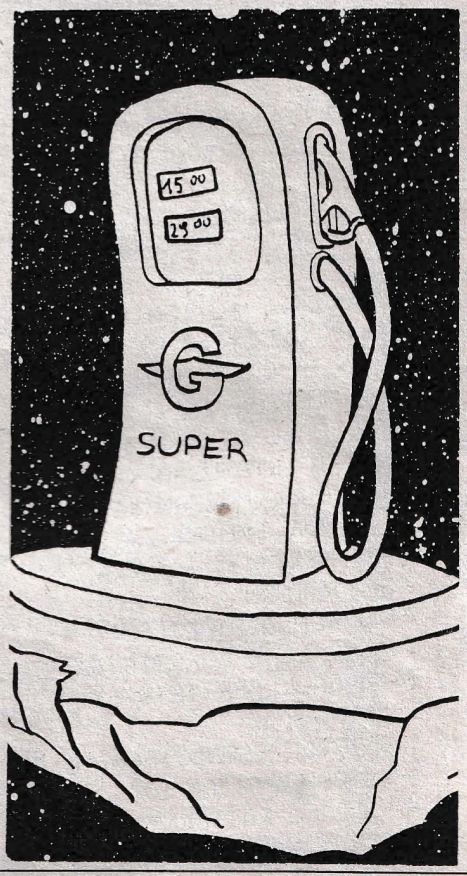
```
3510 FOR J=1 TO 20
3520 NEXT J
3530 CALL PATTERN(#7,32)
3540 FOR J=1 TO 20
3550 NEXT J
3560 NEXT I
3570 LET T=35
3580 LET C=3
3590 CALL MOTION(#1,0,C,#2,0
,C,#3,0,C,#4,0,C)
3600 LET SP=SP+1
3603 LET S=S+1
3610 IF SP>A THEN GOTO 4000
3620 DISPLAY AT(10,2):"NAECH
STER SPIELER...(ENTER)"
3630 CALL KEY(3,KEY,STATUS)
3640 IF STATUS=0 THEN 3630
3650 IF KEY<>13 THEN 1210
3660 DISPLAY AT(10,1):" "
3670 CALL DELSPRITE(#7)
3680 GOTO 3000
4000 REM SCHLUSS
4010 REM
4020 REM
4030 PRINT "ICH BEDANKE MICH
"
4999 STOP
5000 SUB DEFINITION
5010 REM
5020 REM
5030 REM    MUTTERSCHIFF 1
5040 CALL CHAR(123,"0000F0FC
FF000000")
5050 REM
5060 REM    MUTTERSCHIFF 2
5070 CALL CHAR(124,"003C66C3
C3663C00")
5080 REM
5090 REM    MUTTERSCHIFF 3
5100 CALL CHAR(125,"C0F0FCFF
FFFCF0C0")
5110 REM
5150 REM    LANDEFAEHRE
5160 CALL CHAR(127,"0018183C
3C7E5A42")
5170 REM
5180 REM    EXPLOSION
5190 CALL CHAR(128,"42084118
9C004210")
5200 REM
5210 REM    LANDSCHAFT
5220 CALL CHAR(129,"00000000
18244281")
5230 REM
5999 SUBEND
6000 SUB SPIELANLEITUNG
6002 CALL DELSPRITE(ALL)
6010 PRINT "VON DEM MUTTERSC
HIFF AUS,"
6020 PRINT "WIRD EINE LANDEF
AEHRE"
```

```
6030 PRINT "GESTARTET, DIE A
UF DEM MOND"
6040 PRINT "GESTEIN SAMMELN
UND DIESES"
6050 PRINT "DANN BEI DEM MUT
TERSCHIFF"
6060 PRINT "WIEDER ABLIEFERN
 SOLL."
6070 PRINT "SIE SIND DER KOM
MANDEUR DER"
6080 PRINT "LANDEFAEHRE UND
MUESSEN"
6090 PRINT "DAFUER SORGEN, D
ASS DIE"
6100 PRINT "LANDEFAEHRE UNBE
SCHADET"
6110 PRINT "LANDET."
6120 PRINT "ZU DIESEM ZWECKE
 DUERFEN SIE"
6130 PRINT "IHR TRIEBWERK EI
NSETZEN."
6140 PRINT "ABER IHRE ENERGI
E IST"
6150 PRINT "BEGRENZT."
6160 FOR I=1 TO 4
6170 PRINT
6180 NEXT I
6190 PRINT "ENTER TASTE DRUE
CKEN >>>>>>"
6200 CALL KEY(3,KEY,STATUS)
```

```
6210 IF STATUS=0 THEN 6200
6220 IF KEY<>13 THEN 6200
6230 CALL CLEAR
6231 PRINT "HABEN  SIE   ES
  GESCHAFFT,ORDNUNGSGEMAESS
 ZU  LANDEN,SO MUESSEN
SIE  NUN  WIEDERSTARTEN UND
IHR MUTTERSCHIFFERREREICHEN.
"
6232 PRINT :: PRINT "DAS IST
 WOHL DIE HAUPT-     SCHWIER
IGKEIT",,,,,,"ENTER TAST
E DRUECKEN >>>>>>>"
6233 CALL KEY(3,KEY,STATUS)
6234 IF STATUS=0 THEN 6233
6235 IF KEY<>13 THEN 6233
6236 CALL CLEAR
6240 DISPLAY AT(5,9):"ZUM GA
SGEBEN"
6250 DISPLAY AT(9,6):">>>
R - TASTE <<<"
6260 DISPLAY AT(12,11):"DRUE
CKEN"
6265 PRINT "ENTER-TASTE DRUE
CKEN >>>>>>"
6270 CALL KEY(3,KEY,STATUS)
6280 IF STATUS=0 THEN 6270
6290 IF KEY<>13 THEN 6270
6300 CALL CLEAR
6999 SUBEND
```

Deepspace
für alle Ataris

Ein Spiel für Leute mit viel Fingerspitzengefühl



Sie müssen versuchen, mit Ihrer "fliegenden Untertasse" eine Tankstelle anzufliegen, bevor der Treibstoff ausgeht. Allerdings ist die Steuerung der Untertasse etwas träge und das "Prallfeld", welches das Spielfeld überspannt, erleichtert das Anlocken auch nicht gerade. Um die Schwierigkeit noch etwas zu erhöhen, kommt nach jedem geglückten Manöver ein Hindernis hinzu, das es zu umfliegen gilt. Gesteuert wird diese Untertasse mit dem Joystick 0.

Lift Bär

für TI-99

Das Programm ist für den TI 99/4 A mit Extended BASIC und Joystick geschrieben.
Nach dem Eintippen oder Laden wird es mit RUN gestartet. Nach einer Pause von ca. 3 Sekunden wird der Bildschirm gelöscht. Dann belegt der Rechner den Grafiksatz um.
Der Bildschirm wird jetzt schwarz, und das Titelbild erscheint in Multicolor-Großbuchstaben. Dabei wird Song of Joy gespielt.
Sie können mit der Taste „J" eine kurze Spielanleitung abrufen. Bei jeder anderen Taste erhalten Sie keine Spielanleitung.
In der Stadt wird ein Neubau errichtet. Es ist kurz nach Feierabend, und die Maurer haben ihre Arbeit beendet. Sie haben jedoch einen Teil ihres Werkzeuges vergessen und einen kleinen Bären beauftragt das Werkzeug für sie zu holen. Bei dem Werkzeug handelt es sich um 2 Hämmer, 2 Gabelschlüssel und 2 Schraubendreher.
Die Aufgabe hört sich zwar leicht an, ist aber gar nicht so einfach zu lösen, da erstens der Bär nicht weiß wo das Werkzeug versteckt ist, zweitens die einzelnen Etagen des Neubaues nur mittels auf- und abfahrenden Fahrstühlen zu erreichen sind und drittens nur eine befristete Zeit zur Verfügung steht.
Sie können den Bären mit dem Joystick nach rechts und links bewegen. Achten Sie jedoch darauf, daß die Taste „ALPHA LOCK" nicht gedrückt ist, da sonst die Joystickabfrage nicht richtig funktioniert. Zum Betreten eines Liftes bewegen Sie den Bären in die Richtung, in der Sie den Lift betreten wollen, immer dann, wenn der Lift gerade vor dem Bären ist und in die gewünschte Richtung fährt.
So lange der Bär im Lift ist und Sie nicht aussteigen wollen, muß der Joystick in der neutralen Mittelstellung stehen.
Zum Verlassen eines Liftes drücken Sie den Joystick wieder in die gewünschte Richtung.
Der Bär soll ja, wie schon zuvor erwähnt, Werkzeug suchen. Das Werkzeug ist in den 12 Fragezeichen versteckt. Zum Aufheben eines Fragezeichens führen Sie den Bären genau darüber und drücken dann den Aktionsknopf am Joystick.
Wenn sich unter dem Fragezeichen ein Werkzeug befindet, so ertönt ein Multiklang-Gong, und das Werkzeug erscheint in der oberen linken Bildschirmecke. Befindet sich unter dem aufgenommenen Fragezeichen jedoch kein Werkzeug, so werden Sie ein Kratzen vernehmen und es kann sein, daß die Wände des Neubaues versetzt werden. Dies kann Ihnen einen günstigeren Weg verschaffen, es kann Ihnen jedoch auch den Weg versperren, so daß Sie einen Umweg zum nächsten Fragezeichen wählen müssen.
Wenn Sie alle 6 Werkzeuge gefunden haben, so bekommen Sie zu Ihren Punkten den verbliebenen Zeitbonus gutgeschrieben, und es geht mit der nächsten Runde weiter.
Insgesamt wird der Schwierigkeitsgrad 7mal erhöht. Danach bleibt die Schwierigkeit konstant.
Für jeden gefundenen Hammer bekommen Sie 80 Punkte, der Gabelschlüssel bringt 100 Punkte und der Schraubendreher sogar 120 Punkte. Für alle 5000 erreichten Punkte erhalten Sie ein Bären-Bonusleben gutgeschrieben. Der Bär verliert sein Leben wenn er
1. zu hoch oder zu tief aus dem Lift springt,
2. gegen eine vertikale Mauer läuft,
3. mit dem Lift versucht durch eine Mauer zu fahren,
4. den Bildschirm nach irgend einer Seite verläßt,
5. wenn der Zeitbonus auf 0 abgelaufen ist.
Die Fahrstühle fahren immer auf und ab. Bei jedem Richtungswechsel wird der anfänglich 3500 Punkte zählende Zeitbonus um 50 Punkte verringert. Mit steigendem Schwierigkeitsgrad fahren die Lifte immer schneller auf und ab.
Somit läuft auch der Zeitbonus immer schneller, und das Ein- und Aussteigen wird immer schwieriger. Nachfolgend ist das Programm in groben Zügen beschrieben. Den exakten und strukturierten Programmaufbau können erfahrene Programmierer der beigefügten Zuordnungstafel entnehmen.

```
100 !@P-
110 REM ****************************
120 REM *    L I F T  -  B A E R   *
130 REM *--------------------------*
140 REM * Ein Spiel fuer TI 99/4 A *
150 REM * in TI Extended BASIC     *
160 REM * mit Joystick zu steuern  *
170 REM *--------------------------*
180 REM * (c) 1984 by Volker Becker *
190 REM *     Steinbacher Str. 10   *
200 REM *     6370 Oberursel 6      *
210 REM ****************************
220 REM
230 REM
240 !@P+
250 OPTION BASE 1
260 DIM VFELD(6),XFELD(9),SF(4),WFELD(3)
270 GOTO.310 :: A :: B :: BER :: BONUS :: BV :: C :: F :: I :: K :: MOT :: P ::
RUNDE :: S :: ST :: SVAR :: T :: V :: VWERT :: X :: X1 :: X2 :: Y :: Z
280 CALL ASCIISET :: CALL BAER :: CALL CLEAR :: CALL COINC :: CALL COLOR :: CALL
 DELSPRITE :: CALL EINLEITUNG :: CALL GCHAR :: CALL GONG :: CALL HCHAR
290 CALL JOYST :: CALL KEY :: CALL LOCATE :: CALL MAGNIFY :: CALL MELODY :: CALL
 MOTION :: CALL PATTERN :: CALL POSITION :: CALL PUNKTE :: CALL SCREEN
300 CALL SONGOFJOY :: CALL SOUND :: CALL SPRITE :: CALL TITEL :: CALL VCHAR :: C
ALL ZEIT :: !@P-
310 CALL CLEAR :: CALL COLOR(5,1,16,6,1,16,7,1,16,8,1,16):: CALL COLOR(3,16,1,4,
16,1,2,14,1,9,6,16)
```

```
320 RANDOMIZE
330 CALL CLEAR :: CALL ASCIISET
340 CALL CLEAR :: CALL TITEL(XFELD(),5,SF(),0,2):: CALL CLEAR
350 CALL KEY(5,T,S):: IF T=74 OR T=106 THEN CALL EINLEITUNG
360 CALL CLEAR
370 CALL SCREEN(2)
380 FOR I=4 TO 20 STEP 4
390 CALL HCHAR(I,4,112,12):: CALL HCHAR(I,18,112,12)
400 NEXT I
410 CALL HCHAR(24,1,113,32)
420 CALL VCHAR(1,1,112,24):: CALL VCHAR(1,32,112,24)
430 !@P+
440 DATA 1,8,120,1,12,128,5,6,128,5,11,136,5,23,120,5,27,136,9,4,120,9,29,128
450 !@P-
460 DATA 13,9,128,13,13,136,13,18,120,13,24,136,17,12,120,17,22,128,17,27,136,21
,4,136,21,15,128,21,28,120
470 DATA 8,2,120,8,16,120,8,30,136,12,2,128,12,16,136,12,30,128,16,2,136,16,16,1
28,20,30,120
480 RESTORE 440
490 FOR I=1 TO 18 :: READ A,B,C :: CALL VCHAR(A,B,C,3):: NEXT I
500 FOR I=1 TO 9 :: READ A,B,C :: CALL HCHAR(A,B,C,2):: NEXT I
510 CALL COLOR(11,7,2,13,7,2,12,1,1,14,1,1,9,6,16):: CALL SCREEN(5)
520 RUNDE=1 :: BER=4 :: Z=3500 :: P=0 :: BONUS=1 :: CALL PUNKTE(P):: CALL ZEIT(Z
):: CALL BAER(BER)
530 CALL MAGNIFY(3)
540 CALL SPRITE(#2,116,13,177,9,#3,116,13,1,121,#4,116,13,177,233)
550 MOT=RUNDE+4 :: CALL MOTION(#2,-MOT,0,#3,MOT,0,#4,-MOT,0)
560 DATA 3,10,3,26,7,9,7,25,11,13,11,23,15,11,15,21,19,8,19,25,23,13,23,22
570 GOSUB 820
580 BV=128
590 CALL SPRITE(#1,124,11,169,41)
600 ON ERROR 730
610 CALL ZEIT(Z)
620 REM Hauptschleife
630 V=V+1 :: CALL POSITION(#3,X,Y):: IF X>200 THEN GOSUB 940
640 CALL JOYST(1,X,Y):: IF X=0 THEN 660 ELSE IF X=-4 THEN CALL PATTERN(#1,124)EL
SE CALL PATTERN(#1,132)
650 CALL SOUND(-5,523,0)
660 CALL MOTION(#1,0,2*X)
670 CALL COINC(ALL,ST):: IF ST THEN 970
680 CALL POSITION(#1,X,Y):: X=(X+7)/8+2 :: Y=(Y)/8+1 :: CALL GCHAR(X,Y,X1)
690 IF X1=BV OR X1=112 OR X1=113 THEN 700 ELSE 730
700 CALL GCHAR(X-2,Y,X1):: CALL GCHAR(X-2,Y+1,X2):: IF X1=BV OR X2=BV THEN 730
710 CALL KEY(1,T,S):: IF T=18 THEN 1080
720 GOTO 620
730 FOR I=1 TO 4 :: CALL MOTION(#I,0,0):: NEXT I
740 CALL PATTERN(#1,140)
750 CALL MELODY
760 BER=BER-1 :: CALL BAER(BER)
770 IF BER<0 THEN CALL DELSPRITE(ALL):: BER=0 :: CALL BAER(BER):: CALL TITEL(XFE
LD(),1,SF(),-1,5):: GOTO 370
780 IF SVAR THEN Z=3500 :: SVAR=0 :: GOTO 1370
790 CALL MOTION(#2,-MOT,0,#3,MOT,0,#4,-MOT,0)
800 SVAR=0
810 GOTO 590
820 RESTORE 560
830 FOR I=1 TO 6
840 VFELD(I)=2
850 NEXT I
860 FOR I=1 TO 12
870 READ A,B
880 F=INT(RND*5+1.4):: IF VFELD(F)=0 THEN 880
890 VFELD(F)=VFELD(F)-1
900 CALL HCHAR(A,B,F+95)
910 NEXT I
920 FOR K=1 TO 3 :: WFELD(K)=0 :: NEXT K
930 RETURN
940 IF V<5 THEN RETURN ELSE V=0 :: MOT=-MOT :: CALL MOTION(#2,-MOT,0,#3,MOT,0,#4
,-MOT,0)
950 Z=Z-50 :: CALL ZEIT(Z):: IF Z<=0 THEN SVAR=-1 :: GOTO 730
960 RETURN
970 FOR I=2 TO 4
980 CALL COINC(#1,#I,16,ST):: IF ST THEN 1010
990 NEXT I
1000 GOTO 720
1010 CALL POSITION(#I,X,Y):: CALL LOCATE(#1,X,Y)
1020 IF I/2-INT(I/2)=0 THEN CALL MOTION(#1,-MOT,0)ELSE CALL MOTION(#1,MOT,0)
1030 CALL POSITION(#1,X,Y):: CALL SOUND(-250,523-X,3):: X=X/8+1 :: Y=Y/8+1 :: CA
```

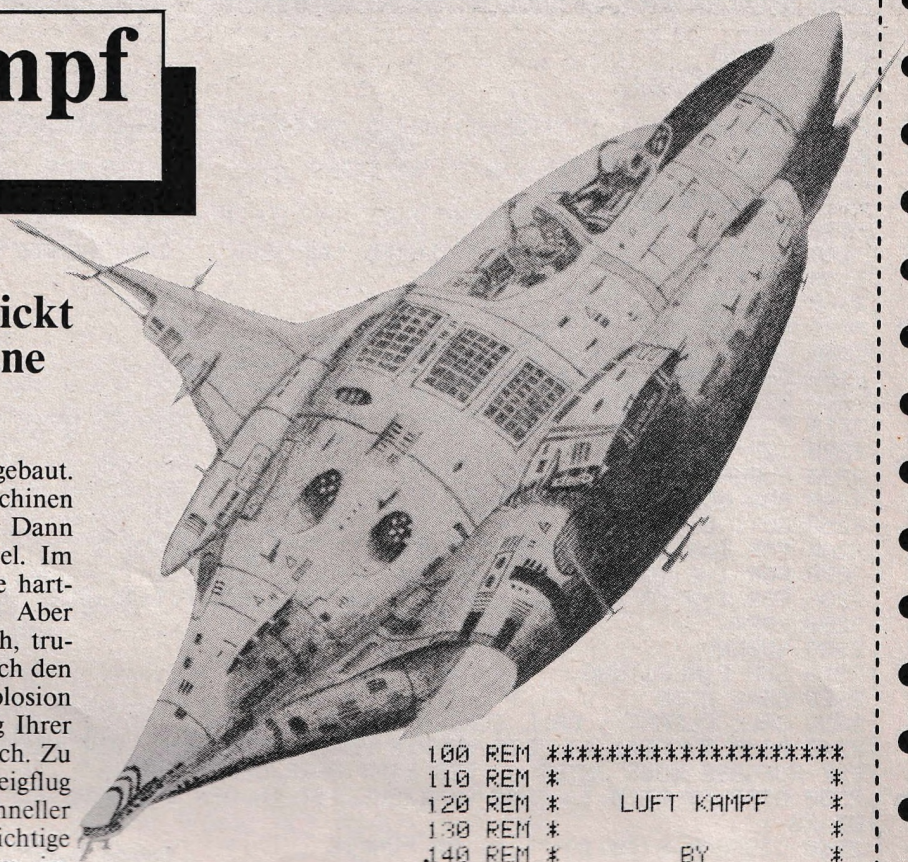
```
LL GCHAR(X,Y,X1):: CALL GCHAR(X,Y+1,X2):: IF X1=BV OR X2=BV THEN 730
1040 CALL JOYST(1,X,Y):: IF X=0 THEN 1030
1050 CALL POSITION(#1,X1,Y)
1060 IF X=-4 THEN CALL LOCATE(#1,X1,Y-18)ELSE CALL LOCATE(#1,X1,Y+18)
1070 GOTO 720
1080 FOR I=1 TO 4 :: CALL MOTION(#I,0,0):: NEXT I
1090 VWERT=0
1100 CALL POSITION(#1,X,Y):: X=(X+7)/8+1 :: Y=(Y)/8+1
1110 CALL GCHAR(X,Y,X1):: CALL GCHAR(X,Y+1,X2)
1120 IF X1<102 AND X1>95 THEN VWERT=X1
1130 IF X2<102 AND X2>95 THEN VWERT=X2
1140 IF VWERT=0 THEN 1170
1150 VWERT=VWERT-95
1160 ON VWERT GOSUB 1190,1210,1220,1230,1240,1250
1170 CALL MOTION(#2,-MOT,0,#3,MOT,0,#4,-MOT,0)
1180 GOTO 720
1190 CALL HCHAR(X,Y,32,2)
1200 CALL COLOR(13,7,2,12,1,1,14,1,1):: CALL SOUND(-500,-7,0):: BV=128 :: RETURN
1210 CALL HCHAR(X,Y,32,2):: CALL COLOR(12,7,2,13,1,1,14,1,1):: CALL SOUND(-500,-
7,0):: BV=120 :: RETURN
1220 CALL HCHAR(X,Y,32,2):: CALL COLOR(14,7,2,13,1,1,12,1,1):: CALL SOUND(-500,-
7,0):: BV=136 :: RETURN
1230 CALL HCHAR(X,Y,32,2):: WFELD(1)=WFELD(1)+1 :: GOSUB 1260 :: RETURN
1240 CALL HCHAR(X,Y,32,2):: WFELD(2)=WFELD(2)+1 :: GOSUB 1260 :: RETURN
1250 CALL HCHAR(X,Y,32,2):: WFELD(3)=WFELD(3)+1 :: GOSUB 1260 :: RETURN
1260 CALL HCHAR(WFELD(VWERT-3),VWERT,VWERT+36)
1270 P=P+20*VWERT :: CALL PUNKTE(P)
1280 IF BONUS*5000<=P THEN BER=BER+1 :: CALL BAER(BER):: BONUS=BONUS+1
1290 CALL GONG
1300 FOR I=1 TO 3
1310 IF WFELD(I)<2 THEN RETURN
1320 NEXT I
1330 RUNDE=RUNDE+1 :: IF RUNDE>8 THEN RUNDE=8
1340 P=P+Z :: CALL PUNKTE(P):: CALL ZEIT(0):: Z=3500
1350 IF BONUS*5000<=P THEN BER=BER+1 :: CALL BAER(BER):: BONUS=BONUS+1
1360 CALL SONGOFJOY(SF())
1370 CALL DELSPRITE(#1):: CALL HCHAR(1,4,32,3):: CALL HCHAR(2,4,32,3)
1380 CALL COLOR(13,7,2,12,1,1,14,1,1)
1390 GOTO 540
1400 !@P+
1410 SUB SONGOFJOY(B())
1420 GOTO 1430 :: AZ :: I :: J :: K :: M1 :: M2 :: STV :: !@P-
1430 DATA 1,195,246,293,391,2,391,493,1,440,523,1,493,587
1440 DATA 1,184,261,293,369,1,493,587,1,440,523,1,391,493,1,329,440
1450 DATA 1,195,246,293,391,2,246,391,1,261,440,1,293,493
1460 DATA 1,184,261,293,369,2,277,493,2,261,440
1470 DATA 1,195,246,293,391,2,391,493,1,440,523,1,493,587
1480 DATA 1,184,261,293,369,1,493,587,1,440,523,1,391,493,1,329,440
1490 DATA 1,195,246,293,391,2,246,391,1,261,440,1,293,493
1500 DATA 1,184,261,293,369,4,261,440,1,195,246,293,391,4,246,391,0
1510 RESTORE 1430
1520 AZ,STV=0
1530 READ K :: IF K=0 THEN SUBEXIT
1540 READ B(1),B(2),B(3),B(4)
1550 FOR I=1 TO K
1560 FOR J=1 TO 4
1570 STV=STV+1 :: IF STV>AZ THEN READ AZ,M1,M2 :: STV=1
1580 CALL SOUND(400,M1,1,M2,0,B(J),5)
1590 NEXT J :: NEXT I
1600 GOTO 1530
1610 !@P+
1620 SUBEND
1630 SUB TITEL(A(),KY,B(),SVAR,FARBE)
1640 GOTO 1650 :: AZ :: I :: J :: K :: M1 :: M2 :: S :: STV :: T :: V :: SPI$ ::
 !@P-
1650 CALL MAGNIFY(2):: IF SVAR=0 THEN CALL CLEAR :: CALL SCREEN(2)
1660 DATA L,I,F,T,B,A,E,R
1670 DATA G,A,M,E,O,V,E,R
1680 CALL HCHAR(6,13,114,8):: CALL HCHAR(7,13,122,8):: CALL HCHAR(8,13,129,8)::
CALL HCHAR(9,13,137,8)
1690 RESTORE 1660
1700 IF SVAR THEN RESTORE 1670
1710 IF SVAR=0 THEN DISPLAY AT(18,3):"SPIELANLEITUNG ? <J : N>"
1720 J=1
1730 FOR K=6 TO 8 STEP 2
1740 FOR I=13 TO 19 STEP 2
1750 READ SPI$ :: CALL SPRITE(#J,ASC(SPI$),FARBE,(K-1)*8+1,(I-1)*8+1)
1760 J=J+1
```

```
1770 NEXT I
1780 NEXT K
1790 DISPLAY AT(24,3)SIZE(23):"@ 1984 BY VOLKER BECKER"
1800 DATA 7,10,11,12,3,13,8,14,5
1810 RESTORE 1800
1820 FOR I=1 TO 9
1830 READ A(I)
1840 NEXT I
1850 CALL COLOR(11,A(1),A(2),12,A(3),A(4),13,A(5),A(6),14,A(7),A(8))
1860 RESTORE 1430
1870 AZ,STV=0
1880 READ K :: IF K=0 THEN 1860
1890 READ B(1),B(2),B(3),B(4)
1900 FOR I=1 TO K
1910 FOR J=1 TO 4
1920 STV=STV+1 :: IF STV>AZ THEN READ AZ,M1,M2 :: STV=1
1930 CALL SOUND(600,M1,1,M2,0,B(J),5):: CALL KEY(KY,T,S):: IF S<>0 THEN 1980
1940 FOR V=9 TO 2 STEP -1 :: A(V)=A(V-1):: NEXT V :: A(1)=A(9)
1950 CALL COLOR(11,A(1),A(2),12,A(3),A(4),13,A(5),A(6),14,A(7),A(8))
1960 NEXT J :: NEXT I
1970 GOTO 1880
1980 CALL DELSPRITE(ALL):: CALL CLEAR
1990 CALL COLOR(11,1,1,12,1,1,13,1,1,14,1,1)
2000 !@P+
2010 SUBEND
2020 SUB PUNKTE(P)
2030 !@P-
2040 DISPLAY AT(1,23)SIZE(5):USING "#####":P :: CALL HCHAR(1,24,38)
2050 !@P+
2060 SUBEND
2070 SUB ZEIT(Z)
2080 !@P-
2090 DISPLAY AT(1,17)SIZE(4):USING "####":Z :: CALL HCHAR(1,18,35)
2100 !@P+
2110 SUBEND
2120 SUB BAER(BER)
2130 !@P-
2140 DISPLAY AT(1,12)SIZE(2):USING "##":BER :: CALL HCHAR(1,13,36)
2150 !@P+
2160 SUBEND
2170 SUB MELODY
2180 GOTO 2190 :: A :: I :: L :: !@P-
2190 DATA 750,195,250,220,500,233,500,195,1000,277
2200 RESTORE 2190
2210 FOR I=1 TO 5 :: READ L,A :: CALL SOUND(L,A,0):: NEXT I
2220 !@P+
2230 SUBEND
2240 SUB GONG
2250 GOTO 2260 :: I :: !@P-
2260 FOR I=0 TO 9 :: CALL SOUND(-150,329,I):: NEXT I
2270 FOR I=0 TO 9 :: CALL SOUND(-150,277,I,329,I+11):: NEXT I
2280 FOR I=0 TO 9 :: CALL SOUND(-150,220,I,277,I+11,329,I+21):: NEXT I
2290 !@P+
2300 SUBEND
2310 SUB EINLEITUNG
2320 GOTO 2330 :: S :: T :: !@P-
2330 CALL CLEAR :: CALL SCREEN(5)
2340 DISPLAY AT(1,1):"DIE MAURER HABEN VERGESSEN  IHR WERKZEUG EIZUSAMMELN UNDDI
ESE AUFGABE MUSS JETZT    DER LIFT BAER UEBERNEHMEN."
2350 DISPLAY AT(5,1):"DAS WERKZEUG IST IN DEN a   VERSTECKT. JE NACH WERKZEUG GI
BT ES :"
2360 DISPLAY AT(8,1):"( =  80 PUNKTE                ) = 100 PUNKTE              *
= 120 PUNKTE"
2370 DISPLAY AT(11,1):"JEDES WERKZEUG IST 2 MAL    VORHANDEN. WIRD JEDOCH EIN  a
 AUFGENOMMEN IN WELCHEM    KEIN WERKZEUG IST SO WERDEN"
2380 DISPLAY AT(15,1):"DIE WAENDE DES NEUBAUES     VERSETZT."
2390 DISPLAY AT(17,1):"LIFT BAER VERLIERT EIN LEBENWENN ER IM FALSCHEN MOMENT  A
US DEM LIFT SPRINGT, EINE  MAUER BERUEHRT ODER DER"
2400 DISPLAY AT(21,1):"ZEITBONUS ABGELAUFEN IST.   BONUSLEBEN ALLE 5000 PUNKTE.N
EUES SPIEL MIT AKTIONSKNOPF"
2410 CALL KEY(1,T,S):: IF T=18 THEN SUBEXIT ELSE 2410
2420 !@P+
2430 SUBEND
2440 SUB ASCIISET
2450 GOTO 2460 :: A$ :: I :: CALL CHAR :: !@P-
2460 DATA 003C66666666663C,001838181818183C,003C66060C18307E,003C66061C06663C,00
0C1C2C4C7E0C0C,007E607C0606663C,001C30607C66663C
```

```
2470 DATA 007E060C1830303,003C66663C66663C,003C66663E060C38,C3BD665E5E66BDC3,FFC
3999981999999,FF83C9C9C3C9C983,FFC3999F9F9F99C3
2480 DATA FF83C9C9C9C9C983,FF819F9F839F9F81,FF819F9F839F9F9F,FFC19F9F919999C3,FF
99999981999999,FFC3E7E7E7E7E7C3,FFF9F9F9F9F999C3
2490 DATA FF9993878F879399,FF9F9F9F9F9F9F81,FFBD9981A5999999,FF99898981919199,FF
81999999999981,FF839999839F9F9F,FFC399999999593C9
2500 DATA FF83999983879399,FFC3999FC3F999C3,FF81E7E7E7E7E7E7,FF999999999999C3,FF
999999C3C3E7E7,FF999999A58199DB,FF99DBC3E7C3DB99
2510 DATA FF99DBC3E7E7E7E7,FF81F9F3E7CF9F81,FEFEFE00EFEFEF,FF18FF18FF18FF18,FFFF
FFFF,0000003C40380478,1F306FD0AFA8ABAB,ABABA8AFA0BF80FF
2520 DATA F80CF60BF515D5D5,D5D515F505FD01FF,FEFEFE00EFEFEF,000000442810101,FFFFF
FFF,018202040202018,000003070D1F0307,0F1F1F0F07040C
2530 DATA 000080C0E0C080C,E0F2F4E4D448C,FEFEFE00EFEFEF,FFFFFFFF,,,00000103070301
03,074F2F272B1203,0000C0E0B0F8C0E,F0F8F8F0E0203
2540 DATA FEFEFE00EFEFEF,FFFFFFFF,,,000000040C1D373F,3F1D08,00000060F2FEF8F8,FAF
EF0680478
2550 RESTORE 2460
2560 FOR I=48 TO 57 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
2570 FOR I=64 TO 90 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
2580 FOR I=112 TO 143 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
2590 FOR I=96 TO 101 :: CALL CHAR(I,"FFC399F9F3E7FFE7"):: NEXT I
2600 CALL CHAR(35,"3C428189919152 3C183418BCBEFE2436"):: CALL CHAR(38,"080C0EFFFF
0E0C08")
2610 CALL CHAR(40,"187EFE7E1818181830189 8F87C0E070340E0703C1E1F0F07")
2620 !@P+
2630 SUBEND
```

Luftkampf

für TI-99



Steuern Sie geschickt Ihre Kampfmaschine

mit Ext. Basic-Modul

Ein Spiel für zwei Personen

Zuerst wird das Spielfeld aufgebaut. Die gegnerischen Kampfmaschinen stehen auf ihren Startrampen. Dann starten Sie in den Nachthimmel. Im Luftkampf müssen Sie nun Ihre hartnäckigen Gegner abschütteln. Aber Vorsicht – kommen Sie zu hoch, trudeln Sie ab, und ein Tiefflug durch den Wald führt unweigerlich zur Explosion der Maschine. Die Handhabung Ihrer Kampfmaschine ist relativ einfach. Zu beachten ist, daß der Jet im Steigflug langsam und im Sturzflug schneller wird. Es liegt nun an Ihnen, die richtige Geschwindigkeit zu finden, um im Luftkampf bestehen zu können.
Gesteuert wird mit Joystick.

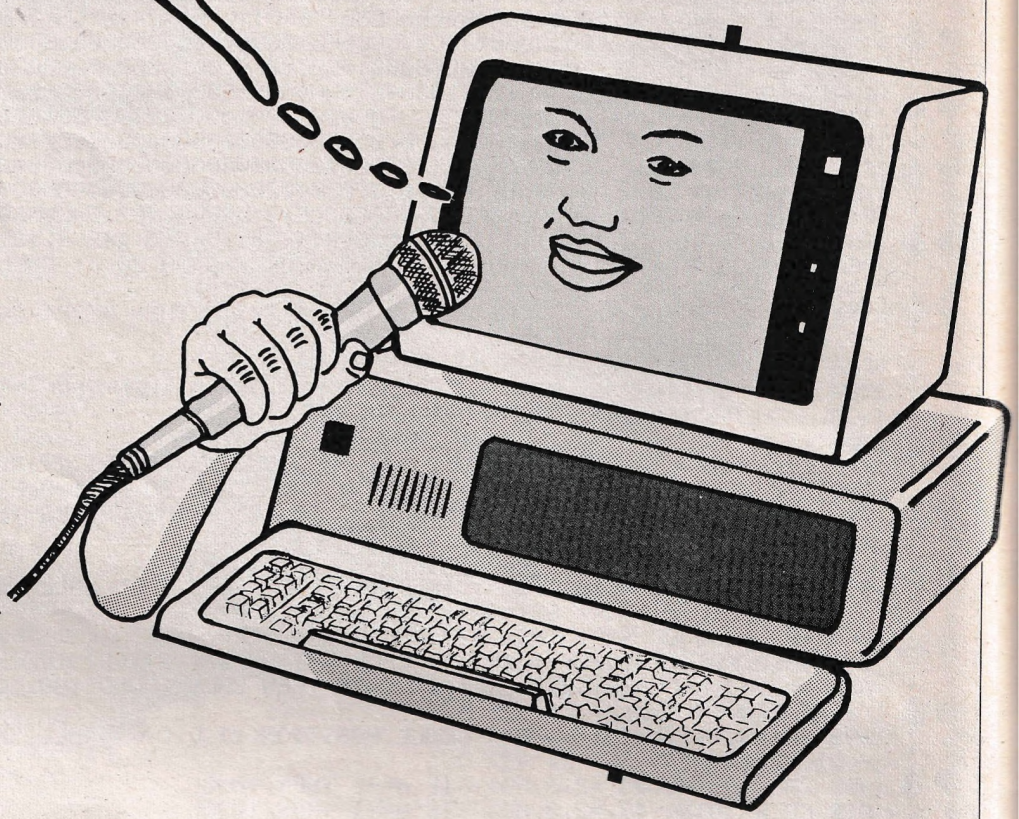
```
100 REM ********************
110 REM *                  *
120 REM *    LUFT KAMPF    *
130 REM *                  *
140 REM *       BY         *
150 REM *                  *
160 REM *    M. STEHLIK    *
220 REM ********************
```

```
250 CALL CLEAR
260 DISPLAY AT(12,1):"     L U F T K A M P F":RPT$("-",28);;;:
270 FOR I=1 TO 16 :: IF INT(I/2)=I/2 THEN CALL SOUND(125,-1,6)ELSE CALL SOUND(12
5,-3,6)
280 NEXT I
290 PRINT "      FUER 2 SPIELER"
300 PRINT :"SIE STARTEN VON IHRER START-";:"RAMPE IN DEN NACHTHIMMEL UM"
310 PRINT :"IHREN GEGNER ZU BEKAEMPFEN.";:"JOYSTICKHEBEL NACH OBEN UND";:
320 PRINT "DAS FLUGZEUG DREHTGEGEN DEN";:"UHRZEIGERSINN, HEBEL NACH";:"UNTEN B
EWIRKT DAS GEGENTEIL.";:
330 PRINT "'ALPHA LOCK' LOESEN UND";:"IRGEND EINE TASTE DRUECKEN."
340 CALL KEY(0,K,S):: IF S=0 THEN 340
```

```
350 SC1=0 :: SC2=0
360 DIM G(16):: FOR I=1 TO 16 :: READ G(I):: NEXT I
370 DATA .2,-.2,-.5,-.7,-1,-.7,-.5,-.2,.2,.4,.6,.8,1,.8,.6,.4
380 CALL CLEAR :: CALL SCREEN(2):: CALL MAGNIFY(3)
390 CALL COLOR(3,16,1,4,16,1,6,16,1,2,7,1,5,13,1)
400 DISPLAY AT(12,1):"            ";SC1;":";SC2
410 DIM X(16):: DIM Y(16)
420 FOR I=0 TO 15 :: X(I+1)=COS(I*PI/8):: Y(I+1)=-SIN(I*PI/8):: NEXT I
430 CALL CHAR(76,"0000000000000000101",78,"0000000000000008080")
440 CALL CHAR(64,"010701070F01070F80E080E0F080E0F801071F3F01073FFF80E0F8FC80E0FC
FF")
450 CALL CHAR(42,"0F0301070101010 1F0C0808080808080")
460 DISPLAY AT(21,1):RPT$("@A",14)
470 DISPLAY AT(22,1):RPT$("BC",14)
480 DISPLAY AT(23,1):RPT$("*+",14)
490 CALL CHAR(40,"FFC1A191898583FFFF83858991A1C1FF")
500 CALL CHAR(68,"FF9FF99FF99FF9FF")
510 CALL HCHAR(24,1,68,32)
520 CALL HCHAR(21,31,41)
530 CALL HCHAR(21,2,40)
540 CALL HCHAR(22,31,40)
550 CALL HCHAR(22,2,41)
560 CALL CHAR(80,"000000000F06070F07060F",82,"0000000000000080F080")
570 CALL HCHAR(23,2,40)
580 CALL HCHAR(23,31,41)
590 CALL CHAR(72,"0000000000004060307030509",74,"000000004050A0C0C0C060")
600 CALL CHAR(84,"0000000000030E07070F06070C"&RPT$("00",9)&"30C080")
610 CALL CHAR(88,"0000000000002040D1F0707010100000000000000000204080802 0C080")
620 CALL CHAR(92,"",93,"0B0F1F120000000000000000404080 80A0E040")
630 CALL CHAR(96,"0000000010101030B0F0F09",98,"0000000000000080A0E0E020")
640 CALL CHAR(100,"0000000002020101050703020000000000000000000000080D0F0F848")
650 CALL CHAR(104,"0000000000040201010403010000000000000000004020B0F8E0E08080")
660 CALL CHAR(108,"0000000000000C0301"&RPT$("0",24)&"C070E0E0F060E030")
670 CALL CHAR(112,"0000000000000000 10F010000000000000000000000F060E0F0E060F0")
680 CALL CHAR(116,"0000000000000001030C000000000000000000020E060F0E0E070C0")
690 CALL CHAR(120,"0000000001030401020400000000000000008080E0E0F8B02040")
700 CALL CHAR(124,"0000000002030705010102020000000000000048F8F0D080")
710 CALL CHAR(128,"0000000004070705010000000000000000000000090F0F0D0C0808080")
720 CALL CHAR(132,"00000000121F0F0B010000000000000000000000040C0E0A080804040")
730 CALL CHAR(136,"0000000101070 71F0D0402000000000000000000080C02080804020")
740 CALL CHAR(140,"0000000C070060F07070E03"&RPT$("0",24)&"80C030")
750 CALL HCHAR(12,1,32,32):: CALL SPRITE(#1,96,10,149,6,#2,96,8,149,238)
760 P1=5 :: P2=5 :: V1=20 :: V2=20 :: FOR I=1 TO 20 :: CALL HCHAR(1+RND*19,1+RND
*31,76):: NEXT I
770 CALL JOYST(1,A,B):: P1=P1+B/4 :: IF P1>16 THEN P1=P1-16 ELSE IF P1<1 THEN P1
=P1+16
780 V1=V1+2*G(P1):: IF V1>20 THEN V1=20
790 CALL POSITION(#1,D,C):: IF D>149 AND D<195 THEN CALL MOTION(#1,0,0,#2,0,0)::
 CALL COLOR(#1,16):: SC2=SC2+1 :: GOTO 1090
800 IF D>195 OR V1<1 THEN P1=13
810 CALL PATTERN(#1,76+4*P1):: CALL MOTION(#1,V1*Y(P1),V1*X(P1))
820 T=2 :: CALL KEY(1,K,S):: IF K=18 THEN CALL SOUND(100,-1,7,1047,7):: GOTO 890
830 CALL JOYST(2,A,B):: P2=P2+B/4 :: IF P2>16 THEN P2=P2-16 ELSE IF P2<1 THEN P2
=P2+16
840 V2=V2+2*G(P2):: IF V2>20 THEN V2=20
850 CALL POSITION(#2,D,C):: IF D>149 AND D<195 THEN CALL MOTION(#1,0,0,#2,0,0)::
 CALL COLOR(#2,16):: SC1=SC1+1 :: GOTO 1090
860 IF D>195 OR V2<1 THEN P2=13
870 CALL PATTERN(#2,76+4*P2):: CALL MOTION(#2,V2*Y(P2),V2*X(P2))
880 T=1 :: CALL KEY(2,K,S):: IF K=18 THEN CALL SOUND(100,1047,7,-1,7):: GOTO 990
 ELSE 770
890 SC1=SC1+1 :: V1=V1-2 :: CALL POSITION(#1,D,C):: CALL SPRITE(#3,76,16,D,C,80*
Y(P1),80*X(P1))
900 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
910 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
920 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
930 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
940 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
950 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
960 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
970 CALL COINC(#3,#2,3,H):: IF H=-1 THEN 1090
980 CALL DELSPRITE(#3):: SC1=SC1-1 :: GOTO 830
990 SC2=SC2+1 :: V2=V2-2 :: CALL POSITION(#2,D,C):: CALL SPRITE(#3,76,16,D,C,80*
Y(P2),80*X(P2))
1000 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1010 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1020 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1030 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
```

```
1040 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1050 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1060 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1070 CALL COINC(#3,#1,3,H):: IF H=-1 THEN 1090
1080 CALL DELSPRITE(#3):: SC2=SC2-1 :: GOTO 770
1090 CALL SOUND(500,-7,0,110,0,111,0):: CALL DELSPRITE(#3)
1100 CALL PATTERN(#T,72):: CALL SOUND(500,-6,2,110,2,117,2)
1110 CALL CHAR(72,"0000000000408050A0A",70,"00000000402010A0500090"):: CALL SOU
ND(500,-5,4,110,4,117,4)
1120 CALL DELSPRITE(ALL):: GOTO 380
```

Lassen Sie Ihren PC zu Wort kommen



Die Firma appli-data GmbH hat die Sprach-Ein-/Ausgabe-Platine PA 600 für den IBM-PC und kompatible PCs entwickelt.
Sie digitalisiert die Sprache, speichert sie ab und gibt sie bei Bedarf über einen Lautsprecher aus.
Durch diese Möglichkeit wird dem PC die Fähigkeit gegeben, Anwendersprache aufzuzeichnen und diese bei Bedarf wieder auszugeben.

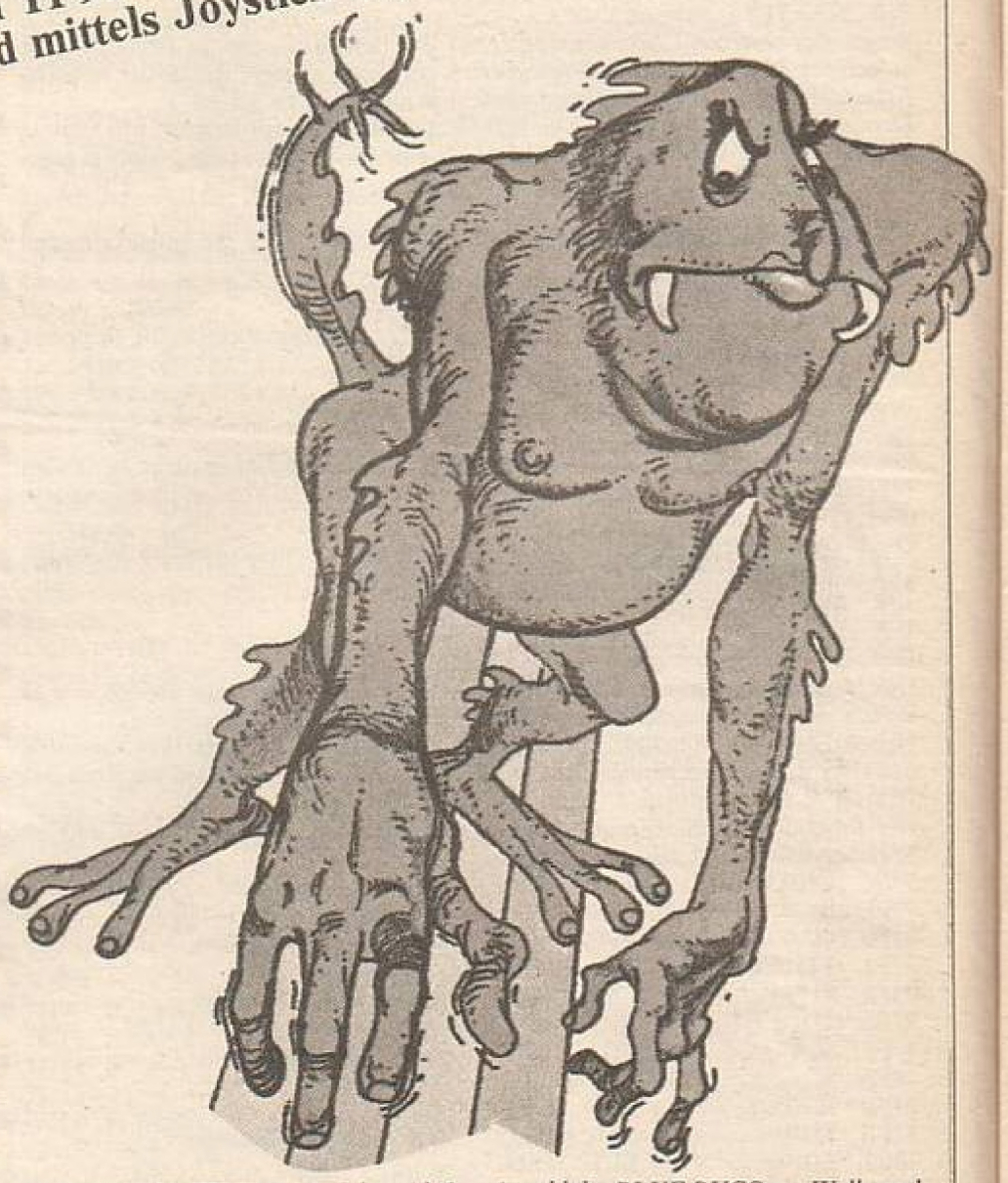
Die Leistungsmerkmale:
- ausgezeichnete Sprachqualität, natürlicher Klang
- Wortschatz über direkte Mikrophon-Eingabe
- automatische Textaufzeichnung auf Diskette oder Platte
- einfacher Einbau in den PC durch Einstecken in einen freien Slot
- Sprachqualität einstellbar durch mitgelieferte Software
- entspricht dem PC Standard
- volle Anwendungsmöglichkeit unter BASIC
- Dokumentation und Bedienerführung in Deutsch

Diese Baugruppe bietet eine Fülle von neuen Anwendungsmöglichkeiten wie zum Beispiel:
- Bedienerführung
- Werbung, Schulung, Training
- Alarm- und Störmeldesysteme
- Automation usw.

BLUE-BUGS

TI 99/4 A

Das Spiel ist auf dem TI 99/4 A in Extended Basic geschrieben und wird mittels Joystick 1 gesteuert.



Weit draußen in einer entfernten Galaxis lebt ein Volk auf dem 2. Mond des Planeten Carylos. Da der Mond keine Atmosphäre hat, leben die Bewohner in kuppelartigen Städten. Eines Tages werden diese friedlichen Bewohner jedoch von blauen, fliegenden Insekten, den „BLUE BUGS" überfallen, welche die Bewohner vernichten wollen. Als einzige Waffe steht eine Torpedokanone zur Verfügung, mit welcher Sie die „BLUE BUGS" vernichten müssen.

Nach dem Eintippen wird das Spiel mit RUN gestartet. Besonderheiten beim Spiel sind nicht zu beachten, da selbst die ‹ALPHA LOCK› Taste eingerastet sein kann. Das Spiel wird Ihnen eine kleine akustische Überraschung servieren. Hierzu sei allerdings nichts gesagt. Lassen Sie sich einfach einmal überraschen.

Nachdem das Titelbild erschienen ist, können Sie mit dem Joystick den Level wählen, in dem Sie beginnen wollen. Der Level (0 bis 9) wird oben links im Bild eingeblendet, und verändert sich, sobald Sie den Joystick bewegen. Haben Sie Ihren Level gewählt, so drücken Sie den Aktionsknopf, um das Spiel zu beginnen.

Die Blue Bugs greifen in Wellen an. Vor jeder Welle wird der augenblickliche Level im Planeten angezeigt. Ein BLUE BUG erscheint mit lautem Getöse auf dem Bildschirm. Dieser BLUE BUG wirft seine Eier ab, die Ihre Laserkanone zerstören können, sobald diese berührt wird. Eine Kanone wird immer dann zerstört, wenn es einem BLUE BUG gelingt bis zum Boden des Mondes vorzudringen, wenn ein BLUE BUG die Kanone berührt, oder wenn diese von den Eiern getroffen wird. Insgesamt stehen 5 Kanonen pro Spiel zur Verfügung.

Nachdem eine Welle vorbei ist, wird der Level erhöht (maximal Level 9), und eine andere Art von BLUE BUGS greift an.

Insgesamt gibt es 3 Arten von BLUE BUGS:

1. Bird BLUE BUGS
2. Spider BLUE BUGS
3. Worm BLUE BUGS

Diese 3 Arten wechseln sich von Welle zu Welle mit ihrem Angriff ab.

Nachdem Sie alle 5 Kanonen verloren haben, erscheint wieder das Titelbild, und Sie können wieder den LEVEL wählen.

Anzahl der BLUE BUGS pro Welle und Punkte pro Treffer:

LEVEL	Anzahl BLUE BUGS	Punkte pro Treffer
0	8	10
1	10	20
2	12	30
3	14	40
4	16	50
5	18	60
6	20	70
7	22	80
8	24	90
9	26	100

Wir wünschen ein Errorfreies Eintippen, und viel Spaß bei ... BLUE BUGS

```
100 !***********************
110 !*  B L U E - B U G S  *
120 !*---------------------*
130 !* TI 99/4 A & X-BASIC *
140 !* Steuerung: Joystick *
150 !*---------------------*
160 !* (c) 1984 Tronicsoft *
170 !* all Rights V.Becker *
180 !***********************
190 !
200 DATA 659,523,659,523,698,587,783,659,659,523,659,523,587,493,523,440,659,523
,659,523,659,523,30000,30000
210 DATA 659,523,659,523,698,587,783,659,659,523,659,523,587,493,523,440,440,349
,440,349,440,349,30000,30000
220 DATA 440,349,440,349,440,349,523,440,587,493,587,493,523,440,440,349,440,349
,440,349,493,391,523,440
230 DATA 493,391,493,391,493,391,493,391,493,391,30000,30000,391,329,493,391,587
,493,587,493,523,440,493,391
240 DATA 440,349,440,349,493,391,523,440,659,523,659,523,659,523,30000,30000,0,0
250 RANDOMIZE
260 OPTION BASE 1
270 DIM C$(4),CF(4),PT$(4),SD(4)
280 DATA 128,132,136,140,293,261,293,329
290 RESTORE 280
300 FOR I=1 TO 4 :: READ CF(I):: NEXT I :: FOR I=1 TO 4 :: READ SD(I):: NEXT I
310 CALL CLEAR :: CALL SCREEN(2):: CALL GRAFIK :: FOR I=0 TO 14 :: CALL COLOR(I,
1,1):: NEXT I
320 CALL BILD :: CALL COLOR(0,2,2,1,11,2,2,16,2,5,9,2,6,13,15,9,11,2,10,11,2,11,
11,2,12,14,8)
330 CALL COLOR(3,13,2,4,13,2)
340 CALL CHPAT(PT$())
350 LV=1 :: CALL TITEL(LV)
360 BUG=0 :: ZV=0 :: P=0 :: GS=5 :: CALL PUNKTE(P):: CALL HCHAR(23,26,121,GS):: 
CALL LEVEL(LV):: RESTORE
370 ZV=ZV+1 :: IF ZV>3 THEN ZV=1
380 ON ZV GOSUB 750,820,890
390 GOSUB 1020 :: CALL COLOR(#2,8)
400 CALL MOTION(#2,LV,RND*30-15)
410 IF STE THEN 430
420 STE=1 :: CALL POSITION(#2,XA,YA):: CALL SPRITE(#4,60,14,XA+16,YA,5+LV,0)
430 FOR LOOP=1 TO 4
440 READ M1,M2 :: IF M1=0 THEN RESTORE :: GOTO 440
450 CALL SOUND(-800,M1,5,M2,5)
460 CALL C1 :: CALL PATTERN(#2,CF(LOOP)):: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,
5*X):: CALL PEEK(-31877,A):: IF (A AND 32)=32 THEN 720
470 CALL KEY(1,T,S):: IF T=18 THEN SKIP=1 :: GOTO 530
480 CALL C2 :: CALL JOYST(1,X,Y):: CALL MOTION(#1,0,5*X):: CALL KEY(1,T,S):: IF
T=18 THEN SKIP=2 :: GOTO 530
490 CALL POSITION(#2,XA,YA):: IF XA>=150 THEN 740
500 CALL POSITION(#4,XA,YA):: IF XA>150 THEN CALL DELSPRITE(#4):: STE=0
510 NEXT LOOP
520 GOTO 400
530 CALL MOTION(#1,0,0):: CALL POSITION(#1,X1,Y1):: CALL SPRITE(#3,124,11,137,Y1
,-128,0)
540 CALL SOUND(-150,-5,0):: CALL POSITION(#2,X2,Y2):: IF ABS(Y1-Y2)<16-LV THEN 5
80
550 CALL DELSPRITE(#3):: READ M1,M2 :: IF M1=0 THEN RESTORE :: GOTO 550
560 CALL SOUND(-800,M1,5,M2,5)
570 ON SKIP GOTO 480,490
580 CALL DELSPRITE(#3):: CALL PATTERN(#2,36):: FOR I=0 TO 5 :: CALL SOUND(-150,-
7,I):: NEXT I
590 CALL PATTERN(#2,40):: FOR I=6 TO 10 :: CALL SOUND(-150,-7,I):: NEXT I
600 CALL PATTERN(#2,92):: FOR I=11 TO 15 :: CALL SOUND(-150,-7,I):: NEXT I :: CA
LL DELSPRITE(#2,#4)
610 P=P+10*LV :: CALL PUNKTE(P):: STE=0 :: BUG=BUG+1 :: IF BUG=6+2*LV THEN 620 E
LSE 390
620 CALL SOUND(-10,30000,0):: LV=LV+1 :: IF LV>10 THEN LV=10
630 CALL LEVEL(LV)
640 BUG=0 :: CALL COLOR(1,INT(RND*13)+3,2,2,INT(RND*13)+3,2):: GOTO 370
650 CALL MOTION(#1,0,0):: CALL PATTERN(#1,80):: FOR I=0 TO 5 :: CALL SOUND(-200,
-5,I):: NEXT I
660 CALL PATTERN(#1,84):: FOR I=6 TO 10 :: CALL SOUND(-200,-5,I):: NEXT I
670 CALL PATTERN(#1,88):: FOR I=11 TO 15 :: CALL SOUND(-150,-5,I):: NEXT I :: CA
LL DELSPRITE(#1)
680 READ M1,M2 :: IF M1=0 THEN RESTORE :: GOTO 680
690 CALL SOUND(-800,M1,5,M2,5)
700 GS=GS-1 :: CALL KANONE(GS):: IF GS=0 THEN 1040
```

```
710 CALL SPRITE(#1,76,16,153,193):: CALL DELSPRITE(#4):: IF STE<>2 THEN STE=0 ::
 GOTO 390 ELSE STE=0 :: GOTO 470
720 CALL COINC(#1,#4,16,ST):: IF ST THEN CALL DELSPRITE(#4):: CALL MOTION(#2,0,R
ND*20-10):: STE=2 :: GOTO 650
730 GOTO 740
740 CALL DELSPRITE(#2,#4):: GOTO 650
750 C$(1)="00003078FC8E060301030102 0C00000000000C1E3F7160C080C080403"
760 CALL SOUND(600,349,0)
770 C$(2)="0000003078FCC6870301020400000000000000C1E3F63E1C080402"
780 CALL SOUND(600,261,0)
790 C$(3)="000000003078FCCE870305000000000000000000C1E3F73E1C0A"
800 CALL SOUND(600,293,0)
810 GOTO 950
820 C$(1)="00003078CD861F3261C68C0C0602000000000C1EB361F84C86633130604"
830 CALL SOUND(600,349,0)
840 C$(2)="0060F0180D063F72C30E1818100000000060F18B060FC4EC370181808"
850 CALL SOUND(600,261,0)
860 C$(3)="C06030180D067FC6830E183000000000003060C18B060FE63C170180C"
870 CALL SOUND(600,293,0)
880 GOTO 950
890 C$(1)="0000000C1E3F73E140C080000000000000000000000183C2E77E3C18"
900 CALL SOUND(600,349,0)
910 C$(2)="000000000C1E3F7321400000000000000000000000000086CCFA783"
920 CALL SOUND(600,261,0)
930 C$(3)="00000000000C1E331120000000000000000000000000008CC8FC3"
940 CALL SOUND(600,293,0)
950 C$(4)=C$(2)
960 CALL SOUND(600,261,0)
970 FOR I=1 TO 4 :: CALL CHAR(4*I+124,C$(I)):: CALL SOUND(300,SD(I),0):: NEXT I
980 CALL SOUND(300,349,0)
990 CALL DELSPRITE(#1):: CALL MAGNIFY(3):: CALL SOUND(300,261,0)
1000 CALL SPRITE(#1,76,16,153,193):: CALL SOUND(300,174,0):: RESTORE
1010 RETURN
1020 CALL SPRITE(#2,132,11,1,1,0,126,#3,132,11,1,1,0,-126)
1030 FOR I=5 TO 0 STEP -1 :: CALL SOUND(-50,-6,I):: NEXT I :: CALL DELSPRITE(#3)
:: CALL MOTION(#2,0,0):: RETURN
1040 CALL DELSPRITE(ALL):: STE=0 :: CALL CHARBACK(PT$( )):: GOTO 350
1050 SUB BILD
1060 DATA 20,10,72,1,21,10,73,1,18,16,72,1,19,16,73,3,18,24,72,1,19,24,73,3,12,2
5,72,1,13,25,73,5
1070 DATA 18,25,74,4,17,26,74,5,19,27,74,3,10,28,72,1,11,28,73,8,19,28,74,3
1080 DATA 14,29,74,8,14,30,74,8,12,31,72,1,13,31,73,1,14,31,74,8,17,32,74,5
1090 DATA A@B,CGGGD,EGGGF,17,7,15,13,9,22
1100 DATA adbef,cbbbbb9,`bbbbbh,bbbbbrs,obbbtvi,nbrsuwj,mP9lk
1110 RESTORE 1060
1120 CALL HCHAR(1,1,31,768)
1130 FOR I=1 TO 100 :: CALL VCHAR(INT(RND*20)+1,INT(RND*31)+1,INT(RND*1.5)+45)::
 NEXT I
1140 FOR I=1 TO 12 :: CALL VCHAR(INT(RND*20)+1,INT(RND*31)+1,33):: NEXT I
1150 FOR I=1 TO 20 :: READ Z,S,C,A :: CALL VCHAR(Z,S,C,A):: NEXT I
1160 READ C1$,C2$,C3$
1170 FOR I=1 TO 3 :: READ Z,S
1180 DISPLAY AT(Z,S)SIZE(3):C1$
1190 DISPLAY AT(Z+1,S-1)SIZE(5):C2$
1200 DISPLAY AT(Z+2,S-1)SIZE(5):C3$
1210 NEXT I
1220 CALL HCHAR(22,1,120,96)
1230 READ C1$ :: DISPLAY AT(4,4)SIZE(5):C1$
1240 FOR I=5 TO 9 :: READ C1$ :: DISPLAY AT(I,3)SIZE(7):C1$ :: NEXT I
1250 READ C1$ :: DISPLAY AT(10,4)SIZE(5):C1$
1260 SUBEND
1270 SUB C1 :: CALL COLOR(5,9,2):: SUBEND
1280 SUB C2 :: CALL COLOR(5,10,2):: SUBEND
1290 SUB STELL(LV):: CALL JOYST(1,X,Y):: IF X OR Y THEN LV=LV+1
1300 IF LV>10 THEN LV=1
1310 DISPLAY AT(1,1)SIZE(1):USING "#":LV-1
1320 SUBEND
1330 SUB LEVEL(LV):: CALL DELSPRITE(ALL):: CALL MAGNIFY(2):: CALL SPRITE(#1,47+L
V,2,33,49):: SUBEND
1340 SUB PUNKTE(P):: DISPLAY AT(1,22)SIZE(7):USING "#######":P :: SUBEND
1350 SUB KANONE(GS)
1360 CALL HCHAR(23,26+GS,120)
1370 SUBEND
1380 SUB CHPAT(PT$( ))
1390 FOR I=1 TO 4 :: PT$(I)="" :: FOR K=0 TO 3 :: CALL CHARPAT(124+I*4+K,A$):: P
T$(I)=PT$(I)&A$ :: NEXT K :: NEXT I :: SUBEND
1400 SUB CHARBACK(PT$( ))
```

```
1410 FOR I=1 TO 4 :: CALL CHAR(124+4*I,PT$(I)):: NEXT I :: SUBEND
1420 SUB TITEL(LV)
1430 DATA 128,8,25,129,132,8,25,161,128,5,23,131,132,5,23,163
1440 DATA 136,8,73,105,140,8,73,137,136,14,71,107,140,14,71,139
1450 RESTORE 1430
1460 CALL MAGNIFY(4):: FOR I=1 TO 8 :: READ C,F,Z,S :: CALL SPRITE(#I,C,F,Z,S):: NEXT I
1470 RESTORE
1480 CALL C2 :: CALL KEY(1,T,S):: IF T=18 THEN 1550
1490 READ M1,M2 :: IF M1=0 THEN RESTORE :: GOTO 1490
1500 CALL SOUND(350,M1,5,M2,5):: CALL STELL(LV)
1510 CALL C1 :: CALL KEY(1,T,S):: IF T=18 THEN 1550
1520 READ M1,M2 :: IF M1=0 THEN RESTORE :: GOTO 1490
1530 CALL SOUND(350,M1,5,M2,5):: CALL STELL(LV)
1540 GOTO 1480
1550 CALL DELSPRITE(ALL):: CALL MAGNIFY(3):: CALL SPRITE(#1,76,16,153,193)
1560 DISPLAY AT(1,1)SIZE(1):"!"
1570 SUBEND
1580 SUB GRAFIK
1590 DATA ,000000081C3E1C08,00282828,0028287C287C2828,8140201109C4320F,F10205091
020418,0182840813A4488F
1600 DATA F040201088840281,0040200109C40008,91020009002041,0082040003A44083,8040
001088000201,000000000030102,00001008
1610 DATA 00000000002,000004081020 4,3C66666666666663C,183878181818183C,3C6666060C
18307E,3C66061C1C06663C,6C6C6C6C7E0C0C0C
1620 DATA 7E60607C0606663C,1C3060607C66663C,7E7E060C1830303,3C66667C3C66663C,3C6
666663E060C38,0010101038387C,000030300030102
1630 DATA 010302050502013,7878D4EC783,80C0C0E0E0C08,000C161A3B371E0C,000000007EF
F00FF,0000000000071C71,0000000000E0388E
1640 DATA 010307070F1F3C3D,80C0E0E0F0F8FCFC,7D7D7979F1E3C3FF,FEFEFEFEFFFFFFFF,FF
FFFFFFFFFFFFFF,00003F3F3F3F3F3F,3F3F3F3F3F3F3F3F
1650 DATA FFFFFFFFFFFFFFFF,0044485060504844,0101010101030306,45EDABFABAF6B7F3,80
80808080C0C06,A2B7D55F5D6FEDCF,2882008101100346
1660 DATA 050CAB22BAD0B7F3,144100818090C24,20B7D05B456DE10F,0002008801100246,010
CAA20BAD0B621,004010010090424,2097401B446DE108
1670 DATA 000000000010004,0108A2008200A401,00000000009002,2011400244010008,00400
00008000008,80020009000041,0000040002A40002
1680 DATA 00400010000002,3F3F3F7F7F7F7FFF,00000003071F3F7F,FFFFFFFFFFFFFFFF,0001
0307070F1F1F,011FFFFFFFFFFFFF,80F8FFFFFFFFFFFF
1690 DATA 000000C0E0F8FCFE,0080C0E0E0F0F8F8,FCFCFCFEFEFEFEFF,FFFEFEFEFEFCFCFC,F8
F8F0E0E0C08,FEFCF8E0C,3F0E3B073FFFF88
1700 DATA 7F3F1F0703,1F1F0F07070301,FF7F7F7F7F3F3F3F,FC70DCE0FCFF1F01,FFF0000000
00FFFF,FF8FC7F183CFF1FF,9FCFE9E3F1C7FFFF
1710 DATA FFFFFF8387F09EFC,C3F380FFFFF0FFFF,FFEFC787070F3F7,C33FE1870F1FFFFF,FF0
0FF00FF00FF,FFEFEFEFC7C783FF,0000007C0810207C
1720 DATA 0018202040202018,0101010101030501,03070D18,8080808080C0A08,C0E0B018,3C
62A12121212122,3E21202020A0613F
1730 DATA 0C1211111214181,1038A8A8A8A84582,,0044444444C54638,,00708888F080887,F0
888484848488F
1740 DATA 88848282828284F8,0000000000000042,4242424242424 23C,00000000000000 3E,40
40405E4242423C,000000000000003E,4040403C0202027C
1750 RESTORE 1590
1760 FOR I=32 TO 143 :: READ C$ :: CALL CHAR(I,C$):: NEXT I
1770 SUBEND
```


Human Engineered Software, 150 North Hill Drive, Brisbane, CA 94005
800-227-6703 (in California 800-632-7979) Dept. C20

HesWare

HesWare is a trademark of Human Engineered Software. Facemaker is a trademark of Spinnaker Software. VIC 20 and Commodore 64 are trademarks of Commodore Electronics Ltd. Atari is a registered trademark of Atari, Inc. IBM is a registered trademark of International Business Machines.

© 1983

CIRCLE 133 ON READER SERVICE CARD

Perlentaucher

TI 99/4A

Sind Sie schon einmal Perlentaucher in der Südsee gewesen? Nein? Mit Hilfe eines TI 99/4A, EXTENDED BASIC Modul und Joysticks können Sie es werden.

Nach der Eingabe des Programms starten Sie es mit „RUN", und das Titelbild erscheint zusammen mit einer Melodie.

Ziel des Spieles ist es, einen Perlentaucher auf den Grund des Meeres zu steuern und Perlen aufsammeln lassen. Am Anfang des Spiels steht der Taucher auf einem Boot. Sie müssen nun versuchen, dieses Männchen durch einen Schwarm gefräßiger Fische sicher zu den Perlengründen zu bringen. Dabei müssen Sie darauf achten, daß Sie die Fische nicht berühren und Ihre Sauerstofftanks sich nicht leeren. Geschieht dies trotzdem, verlieren Sie eines Ihrer fünf Leben. Hat der Taucher den Meeresboden erreicht, muß eine Perle aufgenommen und zum Boot zurückgebracht werden. Nachdem die zehnte Perle zum Boot gebracht wurde, beginnt eine neue Runde, die noch schwieriger ist.
Die Steuerung des Spiels erfolgt mit dem Joystick. Der Taucher kann in acht verschiedene Richtungen gesteuert werden. Die Sauerstofftanks können nur auf dem Boot aufgefüllt werden.
Die Punkte verteilen sich wie folgt:
- große Perle 100 Punkte,
- mittlere Perle 50 Punkte,
- kleine Perle 25 Punkte.

Von den kleinen Perlen gibt es 5, von den mittleren 3 und von den großen 2.

Der Bildschirm ist folgenderweise aufgebaut:
Oben links steht die Anzeige für die Punkte, rechts daneben ist die Anzeige für den High-Score. Darunter befindet sich die Anzeige der noch vorhandenen Taucher. Daneben wird angezeigt, ob Sie eine Perle aufgenommen haben. Unter der Luftanzeige befindet sich die Rundenanzeige.

Einige Programmzeilen sind so lang, daß der Computer sie nicht annimmt. In diesem Fall müssen Sie die Zeile mit „enter" beenden und mit „FCTN+REDO" zurückholen. Sie können die Zeile nun ganz normal beenden.

```
100 ! ***********************
110 !************************ **
120 !** (C)22.11.84  BY    **
130 !**      JUERGEN       **
140 !** WOHLFAHRT-MAHN     **
150 !**                    **
160 !************************ **
170 !**                    **
180 !**      TI99/4A       **
190 !** EX.BASIC+JOYSTICK  **
200 !**                    **
210 !************************
220 ! ***********************
```

```
230 !
240 !
250 GOTO 290 :: GE :: RU :: M$ :: I :: PU :: PA :: PE :: LE :: WW :: MM :: HPU :
: LU :: AA :: BB :: CC :: DD :: X :: Y :: K :: S :: C
260 X1 :: Y1 :: BO :: P1 :: P2 :: ZU :: ZZ :: LA :: TON
270 CALL CHAR :: CALL CHARPAT :: CALL COLOR :: CALL SCREEN :: CALL GCHAR :: CALL
 HCHAR :: CALL DELSPRITE :: CALL SOUND :: CALL POSITION :: CALL LOCATE :: CALL P
ATTERN :: CALL MOTION
280 CALL CLEAR :: CALL MAGNIFY :: CALL SPRITE :: CALL KEY :: CALL JOYST :: CALL
COINC
290 !@P-
300 CALL SCREEN(2):: FOR I=2 TO 8 :: CALL COLOR(I,15,2):: NEXT I :: CALL CLEAR :
: RANDOMIZE
310 CALL CHAR(64,"3C4299A1A199423C",136,"3C42B1B18181423C")
320 DISPLAY AT(10,2):"P E R L E N T A U C H E R"
330 DISPLAY AT(12,3):" @1984 BY JUERGEN" :: DISPLAY AT(14,5):"WOHLFAHRT-MAHN"
340 FOR I=1 TO 12 :: CALL SPRITE(#I,136,INT(RND*14)+3,230,I*20,-INT(RND*10),0)::
 NEXT I
350 CALL SOUND(100,440,0):: CALL SOUND(100,550,0):: CALL SOUND(100,660,0):: CALL
 SOUND(200,790,0)
360 GOSUB 1900 :: DISPLAY AT(24,2):"TASTE DRUECKEN"
370 CALL KEY(0,K,S):: IF S=0 THEN 370
380 CALL SOUND(100,790,0):: CALL SOUND(100,660,0):: CALL SOUND(100,550,0):: CALL
 SOUND(200,440,0)
390 CALL CLEAR
400 CALL DELSPRITE(ALL)
410 DISPLAY AT(10,2):"BRAUCHEN SIE EINE ANLEITUNG              J N
          ?"
420 CALL KEY(0,K,S):: IF S=0 THEN 420 ELSE IF K=78 OR K=110 THEN 560
430 CALL CLEAR
440 DISPLAY AT(1,6):"PERLENTAUCHER" :: DISPLAY AT(3,2):"SIE MUESSEN EINEN TAUCHE
R" :: DISPLAY AT(5,2):"DURCH EINEN FISCHSCHWARM ZU"
450 DISPLAY AT(7,2):"PERLEN GRUENDEN STEUERN." :: DISPLAY AT(9,2):"DIESES IST KE
INE LEICHTE" :: DISPLAY AT(11,2):"AUFGABE, DA DIE FISCHE"
460 DISPLAY AT(13,2):"SCHNELL UND GEFRAESSIG SIND" :: DISPLAY AT(24,30):">>"
470 CALL KEY(0,K,S):: IF S=0 THEN 470
480 CALL CLEAR
490 DISPLAY AT(1,6):"PERLENTAUCHER" :: DISPLAY AT(3,2):"STEUERUNG:" :: DISPLAY A
T(5,2):"SIE MUESSEN IHREN TAUCHER,"
500 DISPLAY AT(7,2):"MIT HILFE DES JOYSTICK 1," :: DISPLAY AT(9,2):"UEBER EINE P
ERLE STEUERN" :: DISPLAY AT(11,2):"UND DANN DEN AKTIONSKNOPF"
510 DISPLAY AT(13,2):"DRUECKEN, WENN SIE EINE " :: DISPLAY AT(15,2):"PERLE AUFGE
NOMMEN HABEN," :: DISPLAY AT(17,2):"MUESSEN SIE SIE ZUM BOOT"
520 DISPLAY AT(19,2):"ZURUECK BRINGEN" :: DISPLAY AT(24,30):">>"
530 CALL KEY(0,K,S):: IF S=0 THEN 530
540 CALL CLEAR
550 CALL SOUND(100,440,0):: CALL SOUND(100,550,0):: CALL SOUND(100,660,0):: CALL
 SOUND(200,780,0)
560 !ZEICHEN DEF.
570 CALL CLEAR
580 CALL CHARPAT(70,M$):: CALL CHAR(74,M$):: CALL CHARPAT(65,M$):: CALL CHAR(87,
M$):: CALL CHARPAT(89,M$):: CALL CHAR(81,M$)
590 CALL CHAR(96,"000003020301070B0B0B0302020206000000808080000C0A0A0A0808080800C0
A0"):!MANN STEIGEN
600 CALL CHAR(100,"000602020203 0B0B0B070103020300000C0808080800A0A0A0C0008080800
000"):!MANN SINKEN
610 CALL CHAR(104,"0000000000011F1B000000000000000000000000000048AF0E41A00000000000
000"):!MANN LINKS
620 CALL CHAR(108,"00000000020510F2758000000000000000000000000000080F8D80000000000000
000"):!MANN RECHTS
630 CALL CHAR(112,"000000000000406070706040000000000000000000000078FCF6FEFC780000000
000"):!FISCH 1
640 CALL CHAR(116,"0000000000080D0F0D080000000000000000000000F8FCF6FCF8000000000
000"):!FISCH 2
650 CALL CHAR(120,"00000000001E3F6F7F3F1E000000000000000000000002060E0E060200000000
000"):!FISCH 3
660 CALL CHAR(124,"00000000001F3F6F3F1F000000000000000000000000010B0F0B010000000000
000"):!FISCH 4
670 CALL CHAR(128,"3C7EFFFFFFFF7E3C000000387C7C7C380000000000183C3C180000000000000
000"):!PERLEN
680 CALL CHAR(136,"00FF7F3A1F0F0703E4FFFFFFFFFFFFFFE7FFFFFFFFFFFFFF80FCFCFCFCFCF
0F0"):!BOOT
690 CALL CHAR(140,"0000000000003F640000000000000FFE7000000000000000000000000000000
```

```
000")!BOOT
700 CALL CHAR(32,"FFFFFFFFFFFFFFFF",40,"FFFFFFFFFFFFFFFF",88,"FFFFFFFFFFFFFFFF")
710 CALL CHAR(64,"0808080808080808000000000000000000000000000000000000000000000000
000")!ZEIGER
720 CALL CLEAR
730 !BILDSCHIRM-MASKE
740 CALL COLOR(2,8,8,1,5,1,8,9,1,13,11,5,14,7,8,7,2,1):: CALL SCREEN(6)
750 FOR I=3 TO 7 :: CALL COLOR(I,2,8):: NEXT I
760 RU,PU,PA,PE=0 :: LE=5 :: WW=24 :: MM=128
770 CALL DELSPRITE(ALL):: CALL CLEAR
780 CALL HCHAR(1,1,40,192):: CALL HCHAR(23,1,88,64):: CALL HCHAR(6,15,136)
790 CALL HCHAR(6,16,137):: CALL HCHAR(6,17,138):: CALL HCHAR(6,18,139)
800 CALL HCHAR(5,16,140):: CALL HCHAR(5,17,141)
810 CALL MAGNIFY(3)
820 DISPLAY AT(1,1)SIZE(11):"PUNKTE:";PU :: DISPLAY AT(1,15)SIZE(14):"HI-PUNKTE:
";HPU
830 DISPLAY AT(2,3)SIZE(5):"LUJT:"
840 CALL HCHAR(2,10,88,15)
850 CALL SPRITE(#28,64,16,9,190)
860 CALL SPRITE(#27,96,2,10,230):: DISPLAY AT(3,26)SIZE(2):LE
870 !SPIEL
880 RU=RU+1 :: GE=RU*1.5+3 :: LU=121 :: DISPLAY AT(3,2)SIZE(8):"RUNDE:";RU
890 RESTORE 900
900 DATA 112,16,1,120,6,-1,112,3,1,120,11,-1,112,4,1,120,15,-1,112,3,1
910 FOR I=2 TO 22 STEP 3 :: READ AA,BB,CC :: DD=INT(RND*3):: CALL SPRITE(#I,INT(
RND*2)*4+AA,BB,I*3.7+42,10,0,GE*CC+DD)
920 CALL SPRITE(#I+1,INT(RND*2)*4+AA,BB,I*3.7+42,96,0,GE*CC+DD):: CALL SPRITE(#I
+2,INT(RND*2)*4+AA,BB,I*3.7+42,182,0,GE*CC+DD):: NEXT I
930 CALL SPRITE(#23,124,2,130,200,0,-GE-7)
940 CALL SPRITE(#24,116,2,141,100,0,GE+10)
950 CALL SPRITE(#25,124,2,157,200,0,-GE-13)
960 GOSUB 1440
970 CALL SPRITE(#1,96,2,24,128)
980 CALL SOUND(100,220,0):: CALL SOUND(100,330,0):: CALL SOUND(100,500,0)
990 !SPIEL HAUPTPROGRAMM
1000 WW=24 :: MM=128
1010 CALL JOYST(1,X,Y):: CALL KEY(1,K,S):: LU=LU-1 :: CALL LOCATE(#28,9,LU+69)::
 IF LU<0 THEN 1610
1020 CALL COINC(ALL,C):: IF C THEN 1610
1030 IF K=18 THEN 1330
1040 IF X=-4 THEN 1090 ELSE IF X=4 THEN 1150
1050 IF Y=4 THEN 1210 ELSE IF Y=-4 THEN 1280
1060 CALL LOCATE(#1,WW,MM)
1070 CALL COINC(ALL,C):: IF C THEN 1610
1080 GOTO 1010
1090 !SCHWIMMEN LINKS
1100 CALL POSITION(#1,X1,Y1):: IF X1=48 AND Y=4 OR X1=168 AND Y=-4 OR X1<48 THEN
 1060 ELSE IF Y1=8 THEN 1060
1110 IF X1>169 AND Y=-4 THEN 1060
1120 CALL PATTERN(#1,104)
1130 MM=MM-8 :: WW=WW-2*Y :: CALL SOUND(-1,-1,0)
1140 GOTO 1060
1150 !SCHWIMMEN RECHTS
1160 CALL POSITION(#1,X1,Y1):: IF X1=48 AND Y=4 OR X1=168 AND Y=-4 OR X1<48 THEN
 1060 ELSE IF Y1=240 THEN 1060
1170 IF X1>169 AND Y=-4 THEN 1060
1180 CALL PATTERN(#1,108)
1190 WW=WW-2*Y :: MM=MM+8 :: CALL SOUND(-1,-1,0)
1200 GOTO 1060
1210 !MANN STEIGEN
1220 CALL POSITION(#1,X1,Y1)
1230 IF X1<>48 THEN 1250 ELSE CALL GCHAR(INT(X1/8),INT(Y1/8)+1,B0)
1240 IF B0<>137 AND B0<>138 THEN 1060 ELSE 1800
1250 CALL PATTERN(#1,96)
1260 IF X1<48 THEN 1270 ELSE WW=WW-8 :: CALL SOUND(-1,-1,0)
1270 GOTO 1060
1280 !MANN SINKEN
1290 CALL POSITION(#1,X1,Y1):: IF X1=168 THEN 1060 ELSE IF X1=24 THEN WW=WW+16
1300 CALL PATTERN(#1,100)
1310 WW=WW+8 :: CALL SOUND(-1,-1,0)
1320 GOTO 1060
1330 !PERLEN AUFNAHME
1340 IF PE<>0 THEN 1010
1350 CALL POSITION(#1,X1,Y1):: CALL GCHAR(INT(X1/8)+.75,INT(Y1/8)+.75,P1)
1360 IF P1=32 OR P1=88 OR P1=40 THEN CALL SOUND(-1,770,0):: GOTO 1370 ELSE PE=P1
 :: GOTO 1390
```

```
1370 CALL GCHAR(INT(X1/8)+.75,INT(Y1/8)+1.75,P2)
1380 IF P2=32 OR P2=88 OR P2=40 THEN CALL SOUND(-1,770,0):: GOTO 1060 ELSE PE=P2
 :: CALL HCHAR(INT(X1/8)+.75,INT(Y1/8)+1.75,32)
1390 CALL SOUND(100,220,0):: CALL SOUND(100,330,0):: CALL SOUND(100,440,0)
1400 CALL HCHAR(INT(X1/8)+.75,INT(Y1/8)+.75,32)
1410 CALL SOUND(100,440,0):: CALL SOUND(100,330,0):: CALL SOUND(200,220,0)
1420 CALL HCHAR(3,32,PE)
1430 GOTO 1010
1440 !
1450 FOR I=1 TO 5
1460 ZU=INT(RND*29)+3
1470 CALL GCHAR(22,ZU,ZZ):: IF ZZ<>32 THEN 1460 ELSE CALL HCHAR(22,ZU,130)
1480 NEXT I
1490 FOR I=1 TO 3
1500 ZU=INT(RND*29)+3
1510 CALL GCHAR(22,ZU,ZZ):: IF ZZ<>32 THEN 1500 ELSE CALL HCHAR(22,ZU,129)
1520 NEXT I
1530 CALL SOUND(-200,330,0,550,0,880,0)
1540 FOR I=1 TO 2
1550 ZU=INT(RND*29)+3
1560 CALL GCHAR(22,ZU,ZZ):: IF ZZ<>32 THEN 1550 ELSE CALL HCHAR(22,ZU,128)
1570 NEXT I
1580 CALL SOUND(-200,440,0,660,0,990,0)
1590 PA=10 :: RETURN
1600 !
1610 !LEBEN VERLUST
1620 CALL SOUND(100,220,0):: CALL SOUND(100,175,0):: CALL SOUND(200,-3,0)
1630 LU=121 :: LE=LE-1 :: IF PE<>0 THEN PE=0 :: PA=PA-1 :: CALL HCHAR(3,32,40)
1640 DISPLAY AT(3,26)SIZE(2):LE
1650 IF LE=0 THEN 1700 ELSE IF PA=0 THEN 880
1660 GOTO 970
1670 !
1680 !ENDE
1690 !
1700 IF PU>HPU THEN HPU=PU :: DISPLAY AT(1,25)SIZE(5):HPU
1710 DISPLAY AT(10,4):"PLWQ WGWIN  <Q:N>? "
1720 DISPLAY AT(12,9)BEEP:"GWME OVER"
1730 CALL KEY(0,K,S):: IF S=0 THEN 1750 ELSE IF K=74 OR K=106 OR K=89 OR K=121 T
HEN 740
1740 STOP
1750 FOR I=1 TO 100 :: NEXT I
1760 DISPLAY AT(12,9):"          "
1770 CALL KEY(0,K,S):: IF S=0 THEN 1790 ELSE IF K=74 OR K=106 OR K=89 OR K=121 T
HEN 740
1780 STOP
1790 FOR I=1 TO 100 :: NEXT I :: GOTO 1720
1800 !AUF DEM BOOT
1810 LU=121 :: CALL PATTERN(#1,96)
1820 CALL SOUND(-200,440,0,220,0):: WW=WW-24 :: IF PE=0 THEN 1060
1830 CALL LOCATE(#1,WW,MM):: CALL SOUND(100,330,0,440,0)
1840 IF PE=128 THEN PU=PU+100*RU ELSE IF PE=129 THEN PU=PU+50*RU ELSE IF PE=130
THEN PU=PU+25*RU
1850 CALL SOUND(100,392,0):: CALL SOUND(100,392,0,330,0):: CALL SOUND(100,392,0,
330,0):: CALL SOUND(200,440,0,392,0)
1860 CALL SOUND(200,392,0,330,0)
1870 PE=0 :: PA=PA-1 :: IF PA=0 THEN CALL DELSPRITE(ALL):: PU=PU+150*RU :: DISPL
AY AT(1,8)SIZE(5):PU :: GOTO 880
1880 DISPLAY AT(1,8)SIZE(5):PU :: CALL HCHAR(3,32,40)
1890 GOTO 1060
1900 !MUSIK
1910 RESTORE 1970
1920 READ LA,TON :: IF LA=0 THEN 1950
1930 CALL SOUND(LA,TON,5,TON+5,10)
1940 GOTO 1920
1950 RETURN
1960 !DATA MUSIK
1970 DATA 200,659,200,659,200,659,600,523,200,587,200,587,200,587,600,493
1980 DATA 200,659,200,659,200,659,200,659,200,698,200,698,200,698,200,659
1990 DATA 200,1046,200,1046,200,1046,600,880,200,659,200,659,200,659
2000 DATA 200,659,200,698,200,698,200,698,200,659,200,987,200,987,200,987
2010 DATA 600,830,200,1318,200,1318,200,1174,600,1046,200,1318,200,1318,200,1174
,600,1046
2020 DATA 200,1318,200,1318,200,1174,600,1046,600,880,600,1318
2030 DATA 0,0
2040 REM DAS PROGRAMM IST    9468 BYTES LANG
```

COAL-MINER

TI-99/4A

Kohlen-Hermann und sein Freund, der Erdgnom Ralf

Coal-Miner ist ein grafik-, sound- und farbreiches Spiel für den TI-99/4A in der Grundversion, wobei der Speicherplatz voll ausgenutzt wurde. Bei diesem Spiel geht es darum, möglichst viel Kohle einzusammeln. Das Männchen namens Hermann läßt sich mit den Tasten S, D, E und der Leertaste in die jeweilige Richtung bewegen (Leertaste für nach unten). Die Kohle wird mittels der Taste A aufgeladen. Man muß dann die Kohle zum Lkw bringen, sich auf die Rampe stellen und zum Verladen der Kohle die Taste F drücken. Danach wird die Karre, die im beladenen Zustand blau ist, wieder weiß. Jedoch kann bei allen Aktionen sich der Erdgnom Ralf in den Weg stellen. Um weiter spielen zu können, mußt du ihm die Kohle zu fressen geben. Dies geschieht mit Hilfe der Taste G. Die Funktionen der Tasten A, G lassen sich nur durchführen, wenn man unmittelbar neben dem Gnom bzw. der Kohle steht. Beim Abladen der Kohle (Taste F) muß man am linken Ende der Abladerampe stehen. Jede Berührung mit einem anderen Gegenstand führt zur Beendigung der jeweiligen Spielrunde. Pro Spiel gibt es drei Spielrunden. Bedenke, daß du bei jeder Spielrunde nur 500 Schritte machen darfst. Jeder Schritt zählt 1 Punkt. Für einen vollbeladenen Lkw gibt es 150 Punkte. Außerdem werden beim Wegfahren des Lkw die Schächte mit neuer Kohle gefüllt. Der Lkw verschwindet dann, wenn man die vierte Ladung Kohle auf den Lkw verladen will. Wenn man den Gnom gefüttert hat, bekommt man als Trostpflaster einen Punktebonus.

```
10 PK=1
20 CALL CLEAR
30 CALL SCREEN(11)
40 PRINT "     *** COAL-MINER ***"
50 PRINT
60 PRINT "        Copyright by"
70 PRINT "      Wilhelm Konrad &"
80 PRINT "      Ernst Hoelzenbein": : : :
90 CALL SOUND(400,131,0,165,0)
100 CALL SOUND(200,131,0,165,0)
110 CALL SOUND(200,165,0,196,0)
120 CALL SOUND(400,196,0,247,0)
130 CALL SOUND(400,262,0,330,0)
140 CALL SOUND(800,196,0,247,0)
150 CALL SOUND(400,165,0,196,0)
160 CALL SOUND(400,131,0,165,0)
170 CALL SOUND(400,196,0,247,0)
180 CALL SOUND(200,196,0,247,0)
190 CALL SOUND(200,196,0,247,0)
200 CALL SOUND(400,196,0,247,0)
210 CALL SOUND(400,196,0,247,0)
220 CALL SOUND(1200,196,0,247,0)
230 CALL SOUND(600,196,0,247,0)
240 CALL SOUND(200,196,0,247,0)
250 CALL SOUND(400,165,0,196,0)
260 CALL SOUND(400,196,0,247,0)
270 CALL SOUND(800,262,0,330,0)
280 CALL SOUND(800,196,0,247,0)
290 CALL SOUND(600,220,0,262,0)
300 CALL SOUND(200,196,0,247,0)
310 CALL SOUND(400,175,0,220,0)
320 CALL SOUND(400,220,0,262,0)
330 CALL SOUND(1600,196,0,247,0)
340 CALL CLEAR
350 CALL SCREEN(14)
360 PRINT "****************************"
370 PRINT "*  SPIELREGELN:COAL-MINER  *"
380 PRINT "****************************":
390 PRINT "Du steuerst Hermann mit
den Tasten:": :
400 PRINT "(E) = nach oben"
410 PRINT "(S) = nach links"
420 PRINT "(D) = nach rechts"
430 PRINT "(LEERTASTE) = nach unten"
440 PRINT "(A) = Aufladen der Kohle"
450 PRINT "(F) = Verladen der Kohle"
460 PRINT "(G) = Auslieferung der
Kohle an den Erdgnom": : : :
470 PRINT "<BITTE EINE TASTE DRUECKEN>"
480 CALL KEY(0,KEY,STATUS)
490 IF STATUS=0 THEN 480
500 CALL SOUND(3400,247,0,460,
0,739,0,-2,0)
510 REM ***GRAFIK***
520 CALL CLEAR
530 CALL SCREEN(2)
540 CALL COLOR(9,9,1)
550 CALL COLOR(16,16,1)
560 CALL COLOR(15,15,1)
570 CALL COLOR(10,10,1)
580 CALL COLOR(7,7,1)
590 CALL COLOR(2,2,1)
600 CALL COLOR(13,13,1)
610 CALL COLOR(14,14,1)
620 CALL COLOR(9,9,1)
630 CALL COLOR(16,16,1)
```

```
640 CALL COLOR(4,8,1)
650 CALL COLOR(6,6,10)
660 CALL COLOR(3,3,1)
670 CALL COLOR(5,5,1)
680 CALL COLOR(8,8,1)
690 CALL COLOR(12,12,1)
700 CALL COLOR(11,11,1)
710 REM DACH
720 CALL CHAR(96,"000000000103070F")
730 CALL CHAR(97,"0000000080C0E0F0")
740 CALL CHAR(98,"1F3F7FFFFFFFFFFF")
750 CALL CHAR(99,"F8FCFEFFFFFFFFFF")
760 CALL CHAR(100,"000000000103070F")
770 CALL CHAR(101,"0000000080C0E0F0")
780 REM WAENDE
790 CALL CHAR(112,"0F0F0F0F0F0F0F0F")
800 CALL CHAR(113,"FFFFFFFFFFFFFFFF")
810 CALL CHAR(114,"F0F0F0F0F0F0F0F0")
820 CALL CHAR(149,"7E7A7E7E7E7E7E7E")
830 CALL CHAR(150,"007E4242427E7E72")
840 CALL CHAR(116,"FF111111FF111111")
850 REM FASS
860 CALL CHAR(88,"0000000000FEFEFE")
870 CALL CHAR(89,"00FEFEFE00FEFEFE")
880 REM BAUM
890 CALL CHAR(81,"F0F0F0F0F0F0F0F0")
900 CALL CHAR(129,"FFFFFEFEFCFEFCF8")
910 CALL CHAR(130,"3F7FFFFF3F5F0703")
920 CALL CHAR(131,"0101010FEF5FFFFF")
930 CALL CHAR(132,"5C5CFCFFFDFDFEFA")
940 REM RASEN
950 CALL CHAR(59,"55555555555555FF")
960 REM ERDE
970 CALL CHAR(72,"55896A945B6B3EAA")
980 REM WASSER
990 CALL CHAR(134,"183C7EFFFFFFFFFF")
1000 REM RAMPE
1010 CALL CHAR(127,"FFAA55AA55AA55AA")
1020 CALL CHAR(121,"55AA55AA55AA55FF")
1030 CALL CHAR(125,"3F3F000000000000")
1040 REM LKW
1050 CALL CHAR(145,"003C7E66667E3C00")
1060 CALL CHAR(80,"FFFFFFFF00000000")
1070 CALL CHAR(137,"3F7FFFFFFFFFFFFF")
1080 CALL CHAR(138,"F8F8F8F8F8F8FFFF")
1090 CALL CHAR(139,"3F3F3F3F3F3FFFFF")
1100 CALL CHAR(140,"0F1F3E3C3838F8F8")
1110 CALL CHAR(141,"FFFF3F3F3F3F3F3F")
1120 CALL CHAR(142,"0F0F00FFFFFFFFFF")
1130 CALL CHAR(143,"FFFF00FFFFFFFFFF")
1140 CALL CHAR(94,"0F0F0F0F0F0F0F0F")
1150 REM HERMANN LINKS
1160 A1$="3058F83030585C6A"
1170 A2$="6A7A797A484848D8"
1180 REM KARRE
1190 A3$="81FF81FF81FF6666"
1200 REM HERMANN RECHTS
1210 A4$="30687C3030585C6A"
1220 A5$="6A7A79784848486C"
1230 REM ERDGNOM
1240 CALL CHAR(58,"BD7EDBDBFF2424E7")
1250 REM KOHLE
1260 CALL CHAR(71,"0000183C7E7EFFFF")
1270 REM POSITIONEN
1280 KK=KK+1
1290 CALL SCREEN(2)
1300 SI=1
1310 CALL VCHAR(1,26,96)
1320 CALL VCHAR(1,27,97)
1330 CALL VCHAR(2,26,98)
1340 CALL VCHAR(2,27,99)
1350 CALL VCHAR(2,25,100)
1360 CALL VCHAR(2,28,101)
1370 CALL VCHAR(3,25,112)
1380 CALL VCHAR(3,26,113)
1390 CALL VCHAR(3,28,114)
1400 CALL VCHAR(4,25,112)
1410 CALL HCHAR(4,28,114)
1420 CALL VCHAR(4,26,113)
1430 CALL VCHAR(4,27,149)
1440 CALL VCHAR(3,27,150)
1450 CALL VCHAR(3,26,116)
1460 CALL VCHAR(3,24,88)
1470 CALL VCHAR(4,24,89)
1480 CALL VCHAR(3,21,81,2)
1490 CALL VCHAR(2,21,129)
1500 CALL VCHAR(2,20,130)
1510 CALL VCHAR(1,20,131)
1520 CALL VCHAR(1,21,132)
1530 CALL HCHAR(5,1,59,14)
1540 CALL HCHAR(5,19,59,14)
1550 CALL HCHAR(6,1,72,14)
1560 CALL HCHAR(6,19,72,14)
1570 CALL HCHAR(7,1,72,14)
1580 CALL HCHAR(7,19,72,14)
1590 CALL HCHAR(8,1,72,14)
1600 CALL HCHAR(8,19,72,14)
1610 CALL HCHAR(9,1,72,2)
1620 CALL HCHAR(9,19,72,14)
1630 CALL HCHAR(10,1,72,2)
1640 CALL HCHAR(10,19,72,14)
1650 CALL HCHAR(11,1,72,14)
1660 CALL HCHAR(11,19,72,14)
1670 CALL HCHAR(12,1,72,14)
1680 CALL HCHAR(12,30,72,3)
1690 CALL HCHAR(13,1,72,14)
1700 CALL HCHAR(13,29,72,4)
1710 CALL HCHAR(14,1,72)
1720 CALL HCHAR(14,19,72,14)
1730 CALL HCHAR(15,1,72,2)
1740 CALL HCHAR(15,19,72,14)
1750 CALL HCHAR(16,1,72,14)
1760 CALL HCHAR(16,19,72,14)
1770 CALL HCHAR(17,1,72,4)
1780 CALL HCHAR(17,28,72,5)
1790 CALL HCHAR(18,1,72,5)
1800 CALL HCHAR(18,28,72,5)
1810 CALL HCHAR(19,1,72,14)
1820 CALL HCHAR(19,19,72,14)
1830 CALL HCHAR(20,1,72,14)
1840 CALL HCHAR(20,32,72)
1850 CALL HCHAR(21,1,72,14)
1860 CALL HCHAR(21,31,72,2)
1870 CALL HCHAR(22,1,72,14)
1880 CALL HCHAR(22,19,72,14)
1890 CALL HCHAR(23,1,72,14)
1900 CALL HCHAR(23,19,72,14)
1910 CALL HCHAR(23,15,134,4)
1920 CALL HCHAR(3,13,127,2)
1930 CALL HCHAR(4,13,121,2)
1940 CALL VCHAR(3,12,125)
1950 GOTO 2130
1960 REM POSITION LKW
1970 CALL VCHAR(4,6,145)
1980 CALL VCHAR(4,9,145)
1990 CALL VCHAR(4,10,145)
2000 CALL VCHAR(4,5,80)
2010 CALL VCHAR(4,7,80)
2020 CALL VCHAR(4,8,80)
2030 CALL VCHAR(4,11,80)
2040 CALL VCHAR(3,5,137)
2050 CALL VCHAR(3,6,138)
2060 CALL VCHAR(3,7,139)
2070 CALL VCHAR(2,6,140)
2080 CALL VCHAR(2,7,141)
2090 CALL VCHAR(3,8,142)
2100 CALL HCHAR(3,9,143,3)
2110 RETURN
2120 CALL VCHAR(2,8,94)
2130 GOSUB 1970
2140 CALL VCHAR(10,3,71)
2150 CALL VCHAR(15,3,71)
```

```
2160 CALL VCHAR(18,6,71)
2170 CALL VCHAR(13,28,71)
2180 CALL VCHAR(18,27,71)
2190 CALL VCHAR(21,30,71)
2200 REM ***SPIELBEGINN***
2210 X=13
2220 Y=1
2230 S=152
2240 CALL CHAR(108,A1$)
2250 CALL CHAR(109,A2$)
2260 CALL CHAR(110,A4$)
2270 CALL CHAR(111,A5$)
2280 CALL CHAR(S,A3$)
2290 CALL CHAR(40,A1$)
2300 CALL CHAR(41,A2$)
2310 CALL CHAR(42,A3$)
2320 CALL CHAR(43,A4$)
2330 CALL CHAR(44,A5$)
2340 CALL SOUND(200,389,0,
843,0,367,0,-8,0)
2350 CALL VCHAR(Y,X,110)
2360 CALL VCHAR(Y+1,X,111)
2370 CALL VCHAR(Y+1,X+1,S)
2380 CALL SOUND(10,-5,0)
2390 CALL KEY(0,KEY,STATUS)
2400 IF STATUS=0 THEN 2390
2410 SI=SI+1
2420 IF SI=500 THEN 4580
2430 IF KEY=65 THEN 2450
2440 GOTO 2990
2450 CALL GCHAR(Y+1,X+2,M)
2460 CALL GCHAR(Y+1,X-1,N)
2470 IF N=71 THEN 2500
2480 IF M=71 THEN 2940
2490 GOTO 2990
2500 IF S=64 THEN 4580
2510 RANDOMIZE
2520 XX=INT(RND*10)
2530 IF XX>6 THEN 2560
2540 CALL VCHAR(9,17,58)
2550 CALL SOUND(100,110,0)
2560 RANDOMIZE
2570 J=INT(RND*6)+1
2580 IF J=7 THEN 2600
2590 GOTO 2610
2600 J=3
2610 IF J=1 THEN 2630
2620 GOTO 2660
2630 CALL VCHAR(10,15,58)
2640 CALL SOUND(100,110,0)
2650 GOTO 2890
2660 IF J=2 THEN 2680
2670 GOTO 2710
2680 CALL VCHAR(15,15,58)
2690 CALL SOUND(100,110,0)
2700 GOTO 2890
2710 IF J=3 THEN 2730
2720 GOTO 2760
2730 CALL VCHAR(18,15,58)
2740 CALL SOUND(100,110,0)
2750 GOTO 2890
2760 IF J=4 THEN 2780
2770 GOTO 2810
2780 CALL VCHAR(18,18,58)
2790 CALL SOUND(100,110,0)
2800 GOTO 2890
2810 IF J=5 THEN 2830
2820 GOTO 2860
2830 CALL VCHAR(13,18,58)
2840 CALL SOUND(100,110,0)
2850 GOTO 2890
2860 IF J=6 THEN 2870
2870 CALL VCHAR(21,18,58)
2880 CALL SOUND(100,110,0)
2890 S=64
2900 CALL SOUND(200,1200,0)
2910 CALL CHAR(S,A3$)
2920 CALL VCHAR(Y+1,X-1,32)
2930 GOTO 2990
2940 IF S=64 THEN 4580
2950 S=64
2960 CALL SOUND(200,1200,0)
2970 CALL CHAR(S,A3$)
2980 CALL VCHAR(Y+1,X+2,32)
2990 IF KEY=70 THEN 3010
3000 GOTO 3520
3010 IF S=64 THEN 3030
3020 GOTO 3520
3030 IF Y=1 THEN 3050
3040 GOTO 3520
3050 IF X=12 THEN 3070
3060 GOTO 3520
3070 CALL GCHAR(2,9,U)
3080 IF U=71 THEN 3130
3090 CALL VCHAR(2,9,71)
3100 CALL SOUND(100,500,0)
3110 S=152
3120 GOTO 3520
3130 CALL GCHAR(2,10,U)
3140 IF U=71 THEN 3190
3150 CALL VCHAR(2,10,71)
3160 CALL SOUND(100,400,0)
3170 S=152
3180 GOTO 3520
3190 CALL GCHAR(2,11,U)
3200 IF U=71 THEN 3250
3210 CALL VCHAR(2,11,71)
3220 CALL SOUND(100,300,0)
3230 S=152
3240 GOTO 3520
3250 CALL HCHAR(1,5,32,7)
3260 CALL HCHAR(2,5,32,7)
3270 CALL HCHAR(3,5,32,7)
3280 CALL HCHAR(4,5,32,7)
3290 CALL SOUND(100,131,0)
3300 CALL SOUND(100,131,0)
3310 CALL SOUND(200,110,0,175,0)
3320 CALL SOUND(200,131,0,220,0)
3330 CALL SOUND(100,175,0)
3340 CALL SOUND(100,175,0)
3350 CALL SOUND(200,165,0,196,0)
3360 CALL SOUND(200,196,0,233,0)
3370 CALL SOUND(100,131,0)
3380 CALL SOUND(100,165,0)
3390 CALL SOUND(150,123,0,196,0)
3400 CALL SOUND(100,131,0,220,0)
3410 CALL SOUND(200,123,0,196,0)
3420 CALL SOUND(200,110,0,175,0)
3430 CALL SOUND(300,110,30)
3440 GOSUB 1970
3450 PK=PK+150
3460 CALL VCHAR(10,3,71)
3470 CALL VCHAR(15,3,71)
3480 CALL VCHAR(18,6,71)
3490 CALL VCHAR(13,28,71)
3500 CALL VCHAR(18,27,71)
3510 CALL VCHAR(21,30,71)
3520 IF KEY=83 THEN 3820
3530 IF KEY=71 THEN 3610
3540 IF KEY=68 THEN 3860
3550 IF KEY=69 THEN 3890
3560 IF KEY=32 THEN 4080
3570 IF KEY<>83 THEN 2390
3580 IF KEY<>68 THEN 2390
3590 IF KEY<>69 THEN 2390
3600 IF KEY<>32 THEN 2390
3610 CALL GCHAR(Y+1,X+2,B)
3620 CALL GCHAR(Y+1,X-1,C)
3630 IF S=152 THEN 2390
3640 IF B=58 THEN 3660
3650 GOTO 3690
3660 CALL VCHAR(Y+1,X+2,32)
```

```
3670 S=152
3680 GOTO 3730
3690 IF C=58 THEN 3710
3700 GOTO 3810
3710 CALL VCHAR(Y+1,X-1,32)
3720 S=152
3730 CALL SOUND(175,131,0,262,0,-3,0)
3740 CALL SOUND(150,349,0,698,0,-3,0)
3750 CALL SOUND(100,349,0,698,0,-3,0)
3760 CALL SOUND(150,440,0,880,0,-3,0)
3770 CALL SOUND(100,440,0,880,0,-3,0)
3780 CALL SOUND(200,523,0,1047,0,-3,0)
3790 CALL SOUND(100,440,0,880,0,-3,0)
3800 PK=PK+25
3810 GOTO 2390
3820 X=X-1
3830 GOSUB 4430
3840 CALL VCHAR(Y+1,X+2,42)
3850 GOTO 2380
3860 X=X+1
3870 GOSUB 4270
3880 GOTO 2380
3890 Y=Y-1
3900 IF Y<1 THEN 4060
3910 CALL GCHAR(Y,X,P)
3920 IF P>44 THEN 3940
3930 GOTO 3950
3940 IF P<152 THEN 4580
3950 CALL VCHAR(Y,X,110)
3960 CALL VCHAR(Y+1,X,111)
3970 CALL GCHAR(Y+1,X+1,P)
3980 IF P>44 THEN 4000
3990 GOTO 4010
4000 IF P<152 THEN 4580
4010 CALL VCHAR(Y+1,X+1,S)
4020 CALL VCHAR(Y+2,X,43)
4030 CALL VCHAR(Y+2,X,44)
4040 CALL VCHAR(Y+2,X+1,42)
4050 GOTO 2380
4060 Y=1
4070 GOTO 2380
4080 Y=Y+1
4090 CALL VCHAR(Y,X,110)
4100 CALL GCHAR(Y+1,X,P)
4110 IF P>44 THEN 4130
4120 GOTO 4140
4130 IF P<152 THEN 4580
```

```
4140 CALL VCHAR(Y+1,X,111)
4150 CALL GCHAR(Y+1,X+1,P)
4160 IF P>44 THEN 4180
4170 GOTO 4190
4180 IF P<152 THEN 4580
4190 CALL VCHAR(Y+1,X+1,S)
4200 IF Y=1 THEN 4230
4210 CALL VCHAR(Y-1,X,43)
4220 CALL VCHAR(Y,X+1,42)
4230 GOTO 2380
4240 CALL VCHAR(1,X,40)
4250 CALL VCHAR(2,X+1,42)
4260 GOTO 4230
4270 CALL CHAR(110,A4$)
4280 CALL CHAR(111,A5$)
4290 CALL GCHAR(Y,X,P)
4300 IF P>44 THEN 4320
4310 GOTO 4330
4320 IF P<152 THEN 4580
4330 CALL VCHAR(Y,X,110)
4340 CALL VCHAR(Y+1,X,111)
4350 CALL GCHAR(Y+1,X+1,P)
4360 IF P>44 THEN 4380
4370 GOTO 4390
4380 IF P<152 THEN 4580
4390 CALL VCHAR(Y+1,X+1,S)
4400 CALL VCHAR(Y,X-1,43)
4410 CALL VCHAR(Y+1,X-1,44)
4420 RETURN
4430 CALL CHAR(108,A1$)
4440 CALL CHAR(109,A2$)
4450 CALL GCHAR(Y,X,P)
4460 IF P>44 THEN 4480
4470 GOTO 4490
4480 IF P<152 THEN 4580
4490 CALL VCHAR(Y,X,108)
4500 CALL GCHAR(Y+1,X,P)
4510 IF P>44 THEN 4530
4520 GOTO 4540
4530 IF P<152 THEN 4580
4540 CALL VCHAR(Y+1,X,109)
4550 CALL VCHAR(Y+1,X+1,S)
4560 CALL VCHAR(Y,X+1,43)
4570 RETURN
4580 PRINT : : : : : : : : : :
```

A lifetime warranty. And manufacturing standards that make it almost unnecessary.

Consider this: Every time you take your disk for a little spin, you expose it to drive heat that can sidetrack data. Worse, take it to the point of no return. Maxell's unique jacket construction defies heat of 140°F. And keeps your information on track.

And Maxell runs clean. A unique process impregnates lubricants throughout the oxide layer. Extending media and head life. How good is Gold?

Maxell's the disk that many drive manufacturers trust to put new equipment through its paces. It's that bug-free.

So you can drive a bargain. But in accelerated tests, Maxell floppys lead the industry in error-free performance and durability. Proving that if you can't stand the heat you don't stand a chance.

maxell®
IT'S WORTH IT.

```
4590 FOR ER=600 TO 200 STEP -20
4600 CALL SOUND(10,ER,0)
4610 NEXT ER
4620 CALL SCREEN(10)
4630 PRINT "*** C O A L - M I N E R ***": : :
4640 PRINT "METER   ";SI: : : :
4650 PK=PK+SI
4660 PRINT "SCORE   ";PK: : : : : : : :
4661 CALL HCHAR(20,4,43,3-KK)
4662 CALL HCHAR(21,4,44,3-KK)
4670 CALL VCHAR(1,1,42,23)
4680 CALL VCHAR(1,32,42,23)
4690 CALL HCHAR(1,2,64,30)
4700 CALL HCHAR(23,2,64,30)
4701 CALL SOUND(100,300,0)
4702 CALL SOUND(100,200,0)
4703 CALL SOUND(100,110,0)
4710 FOR MM=1 TO 2500
4720 NEXT MM
4730 CALL CLEAR
4735 IF Y=2 THEN 4737
4736 GOTO 4740
4737 IF KK<3 THEN 4750
4740 IF KK>=3 THEN 4770
4750 GOTO 1280
4770 PRINT : : : : : : : : : : : : : : : : : : :
4771 FOR LL=1 TO 30
4772 CALL SOUND(10,7000,0)
4773 NEXT LL
4780 PRINT "          E N D E": : : : : : : :
: : : : :
4790 PRINT "NEUER START    J ODER N"
4791 CALL SOUND(400,6000,0,131,0,500,0,-2,0)
4800 CALL KEY(0,KEY,STATUS)
4810 IF STATUS=0 THEN 4800
4820 CALL CLEAR
4830 PK=1
4835 KK=0
4840 IF KEY=74 THEN 1280
4850 END
```

TIPS + TRICKS
TI-99

Ausgabe eines Textes durch Farbwechsel

Mit Hilfe der folgenden Routine ist es möglich, größere Texte auf einmal auf dem Bildschirm des *TI-99/4a ohne Extended Basic* erscheinen zu lassen. Dabei wird vermieden, daß der Text von unten durchgeschoben wird. Überdies ist dieses Verfahren auch recht schnell.

```
100 CALL CLEAR
110 FOR I=1 TO 16
120 CALL COLOR (I,1,1)
130 NEXT I
140 Print „TEXAS INSTRUMENTS“
150 FOR I=1 TO 16
160 CALL COLOR (I,2,1)
170 NEXT I
```

Programmbeschreibung:

110–130 Der gesamte Zeichensatz wird auf die Farbe Transparent der Hintergrundfarbe angepaßt.

140 Der Text wird geprintet, ohne daß er gesehen wird.

150–170 Der gesamte Zeichensatz bekommt wieder die alte Vordergrundfarbe Schwarz und die alte Hintergrundfarbe Transparent.

★★★★★★★★★★★★★★

++ Neu +++ Neu +++ Neu +++ Neu +++ Neu +++ Neu +++ Neu +++ Neu +++ Neu ++

Jetzt überall im Zeitschriften-handel:

Das Softwaremagazin

METEOR
TI–99/4A

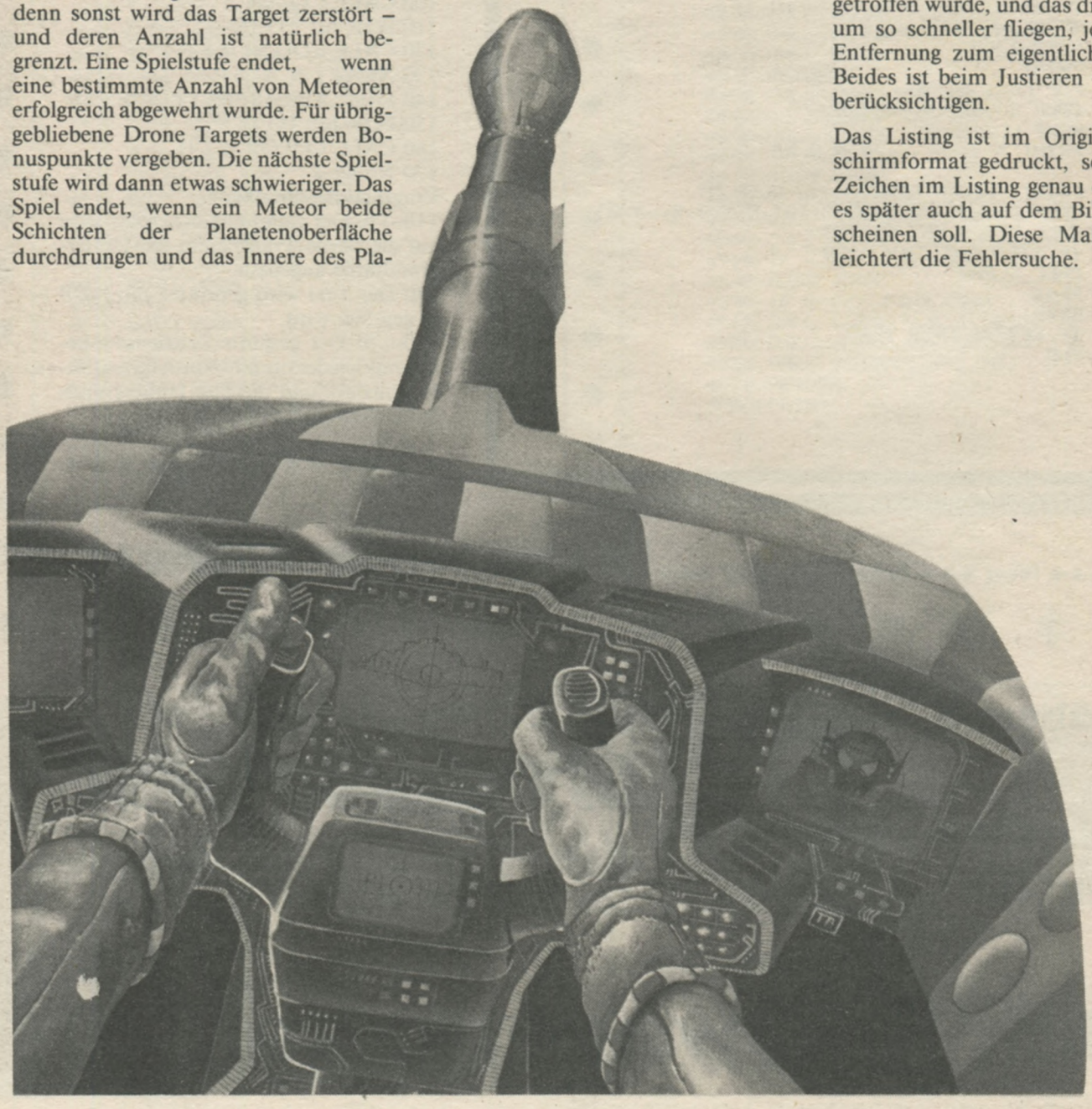
Schützen Sie „Andromeda“ vor Meteoreinschlägen

Zum Schutz des Planeten *„Andromeda“* stehen Ihnen Geschosse zur Verfügung, die mit Hilfe manövrierfähiger Zielscheiben (Drone Targets) auf ihr Ziel gerichtet werden. Sobald sie gestartet wurden, steuern sie geradlinig das Ziel an. Man muß das Target also so geschickt positionieren, daß das Geschoß auf seinem Weg einen Meteor trifft, denn sonst wird das Target zerstört – und deren Anzahl ist natürlich begrenzt. Eine Spielstufe endet, wenn eine bestimmte Anzahl von Meteoren erfolgreich abgewehrt wurde. Für übriggebliebene Drone Targets werden Bonuspunkte vergeben. Die nächste Spielstufe wird dann etwas schwieriger. Das Spiel endet, wenn ein Meteor beide Schichten der Planetenoberfläche durchdrungen und das Innere des Planeten erreicht hat, die Geschoßbasis des Spielers getroffen wurde oder alle Targets verbraucht wurden.
Für den äußersten Notfall gibt es pro Spielrunde einen Superzapper, der einen Meteor sofort zerstört.

Zur Steuerung des Targets werden die Pfeiltasten benutzt. Für die Geschosse und den Superzapper die Tasten „.“ und „,“, wobei die *„Alpha-Lock“*-Taste gedrückt sein muß.

Bleibt noch zu bemerken, daß die Höhe der Punktzahl für einen Treffer davon abhängt, in welcher Höhe der Meteor getroffen wurde, und das die Geschosse um so schneller fliegen, je größer die Entfernung zum eigentlichen Ziel ist. Beides ist beim Justieren des Ziels zu berücksichtigen.

Das Listing ist im Original-TI-Bildschirmformat gedruckt, so daß jedes Zeichen im Listing genau da steht, wo es später auch auf dem Bildschirm erscheinen soll. Diese Maßnahme erleichtert die Fehlersuche.



```
******** METEORS *********
*                         *
*                         *
*                         *
*** (C) BY  BENNO BECKER ***
*                         *
**** BOLDERBERGWEG  1  ****
*                         *
***** 4000 DUESSELDORF *****
*                         *
*                         *
*                         *

1 GOTO 119
2 RANDOMIZE :: STAGE,SZ=1 ::
 MET=10 :: M$="TARGET OUT OF
 FIRING RANGE!" :: CALL CLEA
R
3 ! SCREEN SETTING
4 CALL COLOR(1,16,1,2,16,7,8
,7,1,9,5,1,10,11,1,11,11,1,1
2,15,1)
5 FOR C=3 TO 7 :: CALL COLOR
(C,16,1):: NEXT C
6 CALL SCREEN(2)
7 CALL CHAR(95,"FFFFFFFFFFFF
FFFF")
8 CALL CHAR(35,"00181C3E3E18
")
9 CALL CHAR(36,"113AFC387EA9
2848")
10 CALL CHAR(59,"000001")
11 CALL CHAR(60,"1C3C7EFFFEF
F7F3C")
12 ! STARS
13 FOR C=1 TO 30 :: CALL HCH
AR(23*RND+1,31*RND+1,59):: N
EXT C
14 FOR C=105 TO 111 :: READ
A$ :: CALL CHAR(C,A$):: NEXT
 C
15 DATA E0FFFFFFFFFFFFFF,60F
0F8FFFFFFFFFF,00000087FFFFFF
FF,0000000080F0FEFF,00000000
000000FF,000000070F0F3FFF,00
0000080808 0C0FF
16 FOR C=96 TO 97 :: READ A$
 :: CALL CHAR(C,A$):: NEXT C
17 DATA 00202123277FFFFF,000
484C4E4FEFFFF
18 FOR C=112 TO 120 :: READ
A$ :: CALL CHAR(C,A$):: NEXT
 C
19 DATA 000000000010F8FF,404
040636363E3FF,000000000000F2
FF,00000000010307FF,80808080
80C0F2FF,0002020206060FFF
20 DATA 0000000001070FFF,000
31E7CFCFCFFFF,0103070F1F3FFF
FF
21 CALL HCHAR(24,1,95,32)
22 CALL HCHAR(24,8,43,6):: C
ALL HCHAR(24,22,43,6)
23 FOR C=1 TO 7 :: CALL HCHA
R(23,C,104+C):: NEXT C
24 RESTORE 25 :: FOR C=1 TO
8 :: READ CHR :: CALL HCHAR(
23,C+7,CHR):: NEXT C
25 DATA 109,110,107,107,108,
109,109,109
26 FOR C=16 TO 17 :: CALL HC
HAR(23,C,C+80):: NEXT C
27 FOR C=18 TO 30 :: READ CH
R :: CALL HCHAR(23,C,CHR)::
NEXT C
28 DATA 109,112,109,113,111,
112,114,115,116,117,118,119,
120
29 KR=8
30 DEF ZV=-(170-PV)/2
31 DEF SV=(PH-125)/2
32 GOSUB 77 :: GOSUB 82 :: I
F PL=0 THEN GOSUB 108 :: DIS
PLAY AT(1,1)SIZE(15):"SCORE
:";POINTS
33 DISPLAY AT(1,27):"%" :: D
ISPLAY AT(2,1):"STAGE :";STA
GE
34 ! TARGET CONTROL
35 CALL KEY(0,K,S):: IF S=0
THEN V,H=0
36 IF K=69 THEN V=-20-3*STAG
E
37 IF K=88 THEN V=20+3*STAGE
38 IF K=83 THEN H=-20-3*STAG
E
39 IF K=68 THEN H=20+3*STAGE
40 IF K=46 THEN V,H=0 :: GOT
O 48
41 IF K=44 THEN IF SZ THEN G
OSUB 116
42 IF K=80 THEN PAUSE=1
43 CALL MOTION(#1,V,H):: CAL
L SOUND(-1000,-7,29)
44 !IMPACT WITH GROUND TEST
45 CALL POSITION(#3,X,Y):: I
F X<180 THEN 35 ELSE V,H=0 :
: GOSUB 60 :: GOSUB 82
46 GOTO 35
47 ! FIRE MISSILE
48 CALL MOTION(#1,0,0):: CAL
L POSITION(#1,PV,PH)
49 IF (PV<175)*(PV>10)*(PH<2
55)*(PH>5)THEN CALL MOTION(#
2,ZV-ZV/2,SV-SV/2)ELSE DISPL
AY AT(10,5):M$ :: DISPLAY AT
(11,1):"" :: GOTO 43
50 IF CLR=4 THEN CALL MOTION
(#3,MSV+10,MSH)
51 CALL COLOR(#2,9)
```

```
52 CALL SOUND(-900,110,15,11
2,15,1000,29,-8,10)
53 CALL DISTANCE(#1,#2,D)::
IF D<60 THEN CALL DELSPRITE(
#1,#2):: GOSUB 113 :: GOSUB
77 :: GOSUB 108 :: GOTO 35
54 CALL POSITION(#3,X,Y):: I
F X>180 THEN CALL DELSPRITE(
#1,#2):: GOSUB 60 :: GOSUB 7
7 :: GOSUB 82 :: GOSUB 108 :
: V,H=0 :: GOTO 35
55 CALL DISTANCE(#2,#3,D1)::
 IF D1>85 THEN 53
56 CALL MOTION(#2,0,0):: CAL
L DELSPRITE(#3):: CALL PATTE
RN(#2,36):: FOR C=1 TO 30 ::
 NEXT C :: CALL DELSPRITE(#2
)
57 FOR C=0 TO 30 STEP 4 :: C
ALL SOUND(100,-7,C):: NEXT C
58 CNT=CNT+1 :: POINTS=POINT
S+(190-X)
59 DISPLAY AT(1,8)SIZE(15):P
OINTS :: GOSUB 77 :: GOSUB 8
2 :: GOTO 35
60 ! IMPACT
61 CALL MOTION(#1,0,0)
62 G=Y/8+1 :: IF G>32 THEN C
ALL DELSPRITE(#3):: GOSUB 77
 :: GOSUB 82 :: RETURN
63 CALL DELSPRITE(#3):: CALL
 GCHAR(23,G,GC):: IF GC=32 T
HEN 71
64 IF G<15.5 OR G>17.5 THEN
67
65 FOR C=3 TO 10 :: CALL COL
OR(9,C,C+3):: CALL SOUND(-40
0,100*C,0,1000*C,0,1200*C,0)
:: NEXT C :: CALL HCHAR(23,1
6,32,2)
66 GOTO 100
67 CALL HCHAR(23,G,32)
68 FOR C=28 TO 5 STEP -2 ::
CALL SOUND(-500,250,30,500,3
0,1000,30,-8,31-C):: NEXT C
69 IF PAT=60 THEN IF (G>=3)*
(G<=30)THEN CALL HCHAR(23,G-
2,32,5):: IF G>13 AND G<20 T
HEN 65 ELSE GOTO 72
70 RETURN
71 CALL GCHAR(24,G,GC):: IF
GC=32 THEN 100
72 CALL HCHAR(24,G,32)
73 FOR C=28 TO 5 STEP -2 ::
CALL SOUND(-500,250,30,500,3
0,1000,30,-8,31-C):: NEXT C
74 IF (G<=3)+(G>=30)THEN RET
URN
75 CALL HCHAR(24,G-2,32,5)
76 FOR C=30 TO 0 STEP -2 ::
CALL SOUND(-500,250,25,252,2
5,(3000+100*C),30,-8,30-C)::
 NEXT C :: RETURN
77 ! NEW MISSILE
78 CALL SPRITE(#2,42,6,190,1
25)
79 FOR C=190 TO 170 STEP -1
80 CALL LOCATE(#2,C,125):: N
EXT C
81 RETURN
82 ! NEW METEOR
83 IF PAUSE THEN CALL KEY(0,
K,S):: IF S=0 THEN 83 ELSE P
AUSE=0
84 MZ=5 :: MP=250*RND+2 :: M
SV=((5+STAGE)*RND)+5+STAGE :
: MSH=(6*RND)-3 :: PAT=35 ::
 CLR=6
85 IF RND<STAGE/10 THEN MZ=R
ND*100 :: PAT=60 :: CLR=15 :
: GOTO 87
86 IF RND<STAGE/6 THEN MZ=RN
D*80+20 :: MSV=5*RND+3 :: MS
H=10*RND-5 :: CLR=4 :: GOTO
89
87 IF MP<100 THEN MSH=MSH+3
88 IF MP>150 THEN MSH=MSH-3
89 IF CNT<MET THEN CALL SPRI
TE(#3,PAT,CLR,MZ,MP,MSV,MSH)
:: RETURN
90 DISPLAY AT(10,3):"ATTACK
WAVE ";STAGE;"COMPLETED" ::
DISPLAY AT(12,5):"BONUS POIN
TS 0"
91 FOR C=8 TO 28
92 CALL GCHAR(24,C,GC):: IF
GC=43 THEN BONUS=BONUS+(100*
STAGE):: DISPLAY AT(12,18):B
ONUS :: CALL SOUND(-10,1000,
0):: CALL HCHAR(24,C,95)
93 FOR DEL=1 TO 30 :: NEXT D
EL :: IF C=14 THEN C=21
94 NEXT C
95 FOR C=1 TO 500 :: NEXT C
:: POINTS=POINTS+BONUS :: BO
NUS=0 :: STAGE=STAGE+1 :: DI
SPLAY AT(10,1):"" :: DISPLAY
 AT(12,1):""
96 DISPLAY AT(1,8):POINTS
97 PL,SZ=1 :: CNT=0 :: MET=1
0+2*STAGE :: GOTO 21
98 RETURN
99 ! END OF GAME
100 CALL DELSPRITE(ALL)
101 FOR C=1 TO 6 :: FOR I=-3
 TO -1 :: CALL SOUND(100,I,0
):: CALL SCREEN(C+I+10):: NE
XT I :: NEXT C
102 FOR C=3 TO 8 :: CALL COL
OR(C,9,1):: NEXT C :: CALL M
```

```
OTION(#3,0,0)
103 DISPLAY AT(10,3):"YOUR G
AME IS OVER!!!" :: DISPLAY A
T(14,3):"PRESS R TO REPLAY"
:: DISPLAY AT(16,3):"PRESS S
 TO STOP"
104 CALL KEY(0,K,S):: IF S=0
 THEN 104
105 IF K=82 THEN CALL CLEAR
:: CALL DELSPRITE(ALL):: RUN
 2
106 IF K=83 THEN CALL CLEAR
:: STOP
107 GOTO 104
108 ! NEW TARGET
109 CALL GCHAR(24,KR,GC):: I
F GC=43 THEN CALL SPRITE(#1,
43,16,180,(KR-1)*8,-80,0)::
CALL HCHAR(24,KR,95):: RETUR
N
110 KR=KR+1 :: IF KR=14 THEN
 KR=22
111 IF KR=28 THEN 100
112 GOTO 109
113 ! SOUND OF TARGET HIT
114 FOR C=0 TO 20 STEP 5 ::
CALL SOUND(-100,-6,C):: NEXT
 C :: RETURN
115 ! SUPERZAPPER
116 CALL MOTION(#1,0,0):: V,
H=0
117 CALL DELSPRITE(#3):: FOR
 C=3 TO 7 :: CALL SCREEN(C):
: CALL SOUND(-100,110,30,110
,30,4000+100*C,29,-8,30-4*C)
:: NEXT C
118 CALL SCREEN(2):: DISPLAY
 AT(1,24):"" :: SZ=0 :: GOSU
B 82 :: RETURN
119 CALL CHAR(43,"0042662418
18")
120 CALL CHAR(42,"0018245A5A
2418")
121 CALL CHAR(37,"E080E02EE2
04080E")
122 CALL CLEAR :: PRINT "PRE
SS 'Y' FOR INSTRUCTIONS  PRE
SS ANY KEY TO START GAME"
123 CALL KEY(0,K,S):: IF S=0
 THEN 123 ELSE IF K<>89 THEN
 2
124 PRINT "DEFEND YOUR PLANE
T AGAINST  FALLING  METEORS!
"
125 PRINT : :"FIRE A MISSILE
 (*) AND IT   WILL HEAD FOR
YOUR DRONE    TARGET (+)."
126 PRINT "MAKE SURE IT WILL
 HIT A     METEOR ON ITS WAY
 AS DRONE  TARGETS HIT BY MI
SSILES ARE LOST!"
127 PRINT :"USE ARROW KEYS T
O MOVE DRONETARGET":"USE '.'
 TO FIRE MISSILES."
128 PRINT "USE ',' TO ACTIVA
TE THE     SUPERZAPPER.":"IT
 WILL DESTROY ANY METEOR - O
NCE PER STAGE."
129 PRINT :"'%'INDICATES ITS
 AVAILABILI-TY.": :
130 INPUT "PRESS 'ENTER' TO
CONTINUE":A$
131 CALL CLEAR
132 PRINT "FOR A PAUSE PRESS
 'P' AND   THE NEXT METEOR W
ILL WAIT   FOR ANY KEY TO BE
 PRESSED   BEFORE IT APPEARS
.": : :
133 PRINT "MAKE SURE THAT 'A
LPHA-LOCK' IS DOWN!"
134 PRINT : : :: INPUT "PRES
S 'ENTER' TO CONTINUE":A$
135 GOTO 2
```


Impressum

Homecomputer
erscheint alle 2 Monate im
Tronic-Verlag, 3444 Wehretal 1

Redaktion:
Axel Gredé (verantwortlich),
Siegfried Görk, Frank Brall, Hartmut Wendt,
Holger Grede, Ottfried Schmidt

Freie Mitarbeiter:
Volker Becker, Rolf Freitag

Gesamtherstellung:
Druckhaus Dierichs Kassel, Frankfurter Str. 168,
3500 Kassel

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel)
sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 0 61 21 / 26 60

Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!

Anzeigenleitung:
Annelie Kratzenberg, Heike Lux

Erscheinungsweise:
Erstkaufstag von
HOMECOMPUTER ist Anfang des Monats

Anzeigenpreisliste:
Bitte Mediaunterlagen anfordern.

Anzeigenannahmeschluß:
Jeweils 3 Wochen vor Erscheinungstermin.

Urheberreicht:
Alle in HOMECOMPUTER veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.
Reproduktionen jeder Art (Fototkopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.

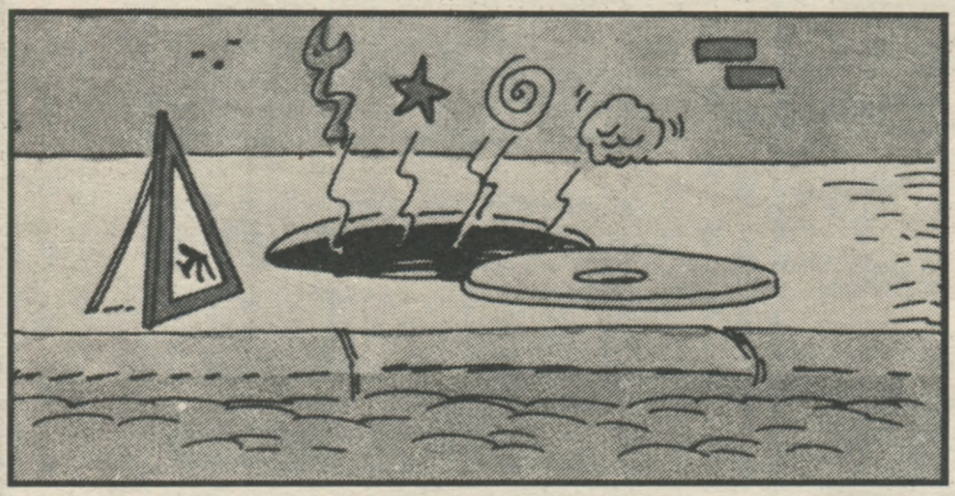
Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreise:
Einzelheft 6,– DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.
Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,– DM pro abgedruckter Seite im Heft einverstanden sind.

Titelblatt:
Werbeatelier H. Kästle, Eschwege

INFRA
TI – 99/4a



Der Abwasserkanal in Ihrer Stadt muß repariert werden. Sie als Kanalarbeiter haben die Aufgabe, in zwei Etagen des Abwassersystems insgesamt 20 Reparaturen vorzunehmen. Sind alle Lecks geschlossen, können Sie wieder an die Oberfläche steigen. Für jede Reparatur benötigen Sie einen Hammer und ein Stück Rohr, die auf den einzelnen Gängen des Abwassersystems liegen. Während der Reparatur treten aber auch kleine Tücken auf, die sich in Form von angreifenden Krokodilen, Giftwolken und Wellen zeigen. Es ist also allergrößte Vorsicht geboten. Auch die Lichtstärke unterhalb der Erde wird ständig geringer, kann jedoch durch Anspringen des Lichtkastens in Höhe der Wasserleitung wieder verstärkt werden. Wenn das Licht vollkommen verloschen ist, ist auch der Kanalarbeiter nicht mehr lebensfähig. Pro Spiel hat man aber drei Leben.

Anzeigen:

HAM: Anzahl der Hämmer
ROHR: Anzahl der Rohre
REPA 1: Reparaturen auf Etage 1
REPA 2: Reparaturen auf Etage 2
Lichtstärkeanzeige ist rechts unten im Bildschirm

Gesteuert wird mit Joystick!

Listing nächste Seite

Hochleistungs-Typenraddrucker im unteren Bereich!

Unter der Vielzahl angebotener Typenraddrucker im unteren Preisbereich handelt es sich bei dem 20 Zeichen/s schnellen DP 20 von Dataproducts um etwas Besonderes. Er bietet dem Anwender bemerkenswerte Vorteile zu einem günstigen Preis.

Er ist besonders anwenderfreundlich konzipiert und neueste ergonomische Erkenntnisse sind im DP 20 verwirklicht. Das Bedienfeld ermöglicht eine vorbildliche optische Kontrolle und bietet den Vorteil, eine Vielzahl von Funktionen über Tastendruck abzurufen.

Zum perfekten Arbeitsplatzdrucker bestimmt ihn das ansprechende Design und die ergonomische Konzeption sowie der erstaunlich niedrige Geräuschpegel von weniger als 55 dB (A).

Mit dem DP 20 gelingt jeder Computerausdruck zu einem lupenreinen Schmuckstück. Dafür steht die sprichwörtliche Dataproducts-Qualität, die auf mehr als 20jährige Erfahrung im Druckermarkt aufbaut.

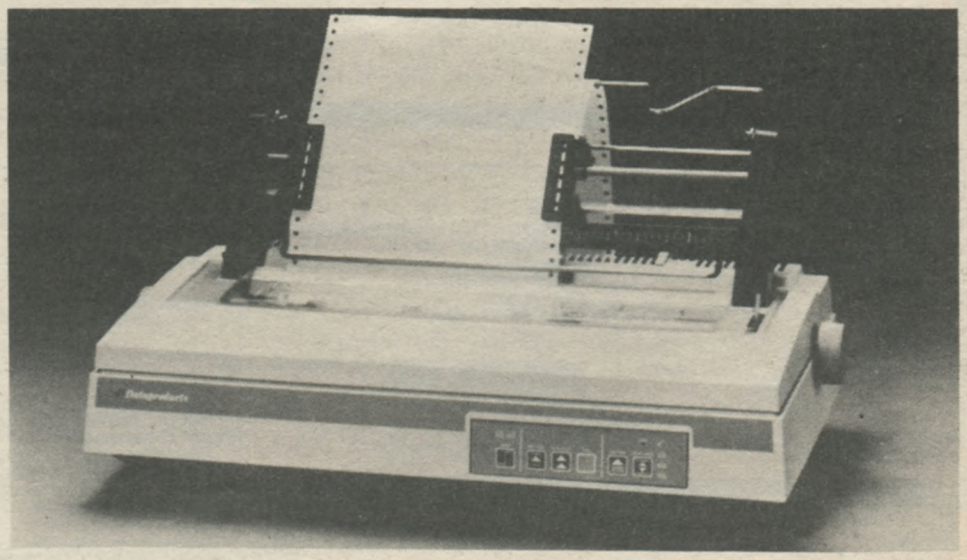
Merkmale und Eigenschaften
Besonders ausführliches Bedienfeld mit Funktionen wie:

★ Formularlängenwahl in mm-Angabe
★ Optische Anzeige von Funktionsarten
★ Opt. Fehlermeldung zur leichten Fehlererkennung

Original plus 3 Kopien, 2 KB Puffer, serielles oder paralleles Interface, Plastiktypenrad, Mylar-Farbbandkassette, Diablo-630 kompatible Textverarbeitung.
Einfache Handhabung bei der Papierzufuhr sowie beim Typenradwechsel, da die wichtigsten Bedienelemente grün gekennzeichnet sind.

Optionen: Bidirektionaler Traktor, automatischer Einzelblatteinzug mit einer und mit zwei Kassetten.

Im Lieferumfang sind Farbband, Plastiktypenrad und Papierführung enthalten. Der Preis für die Grundversion beträgt ca. 2000,– DM zzgl. MwSt.
Dataproducts GmbH, Frankfurter Str. 172–176, 6078 Neu-Isenburg, Tel. 0 61 02 / 30 10.



```
160 ! **INFRA***********
170 DIM L(2,2),RZ(2,2),RH(2,2),SC$(3),SP(3),SF(3),LS(2)
180 CALL CLEAR :: CALL MAGNIFY(3):: CALL SCREEN(6):: GOTO 240
190 M=-500 :: FOR II=1 TO 5 :: RESTORE 230 :: M=M+210
200 FOR I=1 TO 13 :: READ F1 :: CALL SOUND(M,F1,0,F1*2,0,F1*3,0):: CALL KEY(1,KT
,KY):: IF KT=18 THEN 220
210 NEXT I :: NEXT II :: GOTO 190
220 RETURN
230 DATA 330,349,392,440,494,440,494,587,494,494,659,622,587
240 DEF Z=INT(RND*2+1)
250 DATA 11,14,17,25,6,28,5,14,12,20,25,23
260 RESTORE 250 :: FOR I=1 TO 2 :: FOR II=1 TO 2 :: READ D1,D2,D3 :: L(I,II)=D1
:: RZ(I,II)=D2 :: RH(I,II)=D3 :: NEXT II :: NEXT I
270 SP(3),SF(3)=5 :: SP(2)=7 :: SP(1)=3 :: SF(1)=15 :: SF(2)=4
280 EI=0
290 SC$(1)="0C1EFF163E1E8C5C3E3E7EBE1C24246C"&RPT$("0",32):: CALL CHAR(140,SC$(1
))
300 SC$(2)="3078FF687C78313A7C7C7E7D38242436"&RPT$("0",32)
310 SC$(3)="183C7E7E7E3C99BDFF7E7E7E3C2424E7"&RPT$("0",32)
320 CALL CHAR(136,RPT$("0",15)&"3060D1A12356D4CDE0000000000000C0F0F87C444060B8CCE
7")
330 CALL CHAR(132,"00000000000000016396DEF9F8FC3F1F00040E193864E09080400000AAAAF
FFF")
340 DATA 000303060C0603020E190C0603060301E0B05848EC4C5858B060C0E03060C
350 DATA 7EDFDFDFDFDFC37E,00CFE7F3F3E7CF,3F6F47C3C3476F3F,FC7EBEDFDFBE7EFC,0C98C
0E6F6CC980C,603F4F8787473F6
360 DATA 0061F7FFFFFFFFFF,010B6FFFFFFFFFFF,838180E7E7E7E7E7,DFDFDF00FDFDFD,42427
E4242427E42,0,FCFCFCFCFCFCFCFC,F0F0F0F0F0F0F0F
370 DATA C0C0C0C0C0C0C0C,FFFFFFFFFFFFFFFF,FFFFC3C3C3C3FFFF,FFC3C3C3C3C3C3C3
380 FOR I=128 TO 110 STEP -1 :: READ D$ :: CALL CHAR(I,D$):: NEXT I
390 IF EI=0 THEN 1550
400 DATA E07030181C0E0703,C06030180C060301,381C0C06070301,8080C0C0E060703,38181C
0C0F0E0787,C36331190C060301,FFFFFFC0E060703
410 DATA 38181C0C0E060703,8381C1C0E0E070F,00050D150405,F8783C1C0E0E0703,FFFFFF,0
101030307060E0C,00CC52CC52CC,1C1838307060E0C
420 DATA 0101030307060E0C,00E785E726E5,000212725269,1C1838307060E0C,FFFFFF80C0C0
606,303018180C0C0606,FFFFFF0000010307,070E1C1C387070E
430 DATA C0C0800000000103,3C4299A1A199423C,001A237222A2,0301010000808 0C,00206EAE
2826,0001DD9151DD,00001C9D514D,00405CDC508C,0306060C0C18383
440 DATA C0E0E060C0C08001,0103030606 0C0C18,3F3F7F60C0C0808,1C1C3C366666C383,8303
010101
450 DATA 0000808080C0C0C,E0E0E06030303018,18180C0C0C060603,80C0C0607038180C,0E07
030101,00000080C0E0603
460 DATA 381C0C06070301,C0E0F0F8EC667331,38181C0C0E060703,3018180C0C060703
470 CALL CLEAR :: RESTORE 400 :: FOR I=40 TO 86 :: READ D$ :: CALL CHAR(I,D$):: 
NEXT I
480 DATA 1,6,2,6,3,6,6,3,7,5,8,7,9,5,10,3,15,5,16,5,17,6,18,6,19,7,22,7,23,7,24,
4,28,5,29,7
490 DATA 30,5,31,3,32,3
500 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL COLOR(1,6,6,12,5,2,10,15,2,11,9,2):
: FOR I=2 TO 9 :: CALL COLOR(I,9,2):: NEXT I
510 RESTORE 480
520 FOR I=1 TO 21 :: READ S,W :: CALL VCHAR(8-W,S,111,W):: NEXT I
530 FOR I=2 TO 30 STEP 7 :: CALL HCHAR(7,I,110):: NEXT I
540 CALL HCHAR(8,1,116,544):: CALL HCHAR(9,1,118,448)
550 FOR I=15 TO 22 STEP 7 :: FOR II=1 TO 31 STEP 2 :: CALL HCHAR(I,II,121):: CAL
L HCHAR(I,II+1,120):: NEXT II :: NEXT I
560 CALL HCHAR(10,1,126,32):: CALL HCHAR(17,1,126,32)
570 DISPLAY AT(11,8)SIZE(14):"PtTt+.3337=HKt" :: DISPLAY AT(12,8)SIZE(14):"QRU),
/ttt:>ILM"
580 DISPLAY AT(13,8)SIZE(14):"t*B+-0.34;?J3N" :: DISPLAY AT(14,8)SIZE(14):"tt(Ut
2Vt6<GttO"
590 CALL VCHAR(15,5,117,7):: DISPLAY AT(18,9)SIZE(10):"@1589ACDEF" :: GOSUB 190
600 LI=100 :: AS=3 :: R1,R2,RO,HA,PU,QI=0 :: CALL CHARSET
610 CALL HCHAR(11,1,118,128):: CALL VCHAR(15,5,118,7):: CALL HCHAR(15,5,121):: C
ALL HCHAR(17,5,126):: CALL HCHAR(18,1,118,32)
620 CALL CHARSET :: CALL CHAR(61,"0C1E7F1434140C"):: CALL SCREEN(6):: CALL COLOR
(1,6,6,12,5,2,10,15,2,11,9,2)
630 CALL HCHAR(1,25,61,3)
640 DISPLAY AT(23,1):"REPA1:    REPA2:    ROHR:  "
uN:
" :: GOSUB 1470
660 CALL VCHAR(8,11,117,7):: CALL VCHAR(15,20,117,7):: CALL DELSPRITE(ALL)
670 CALL CHAR(140,SC$(1)):: CALL SPRITE(#1,140,11,43,236)
680 FOR I=235 TO 81 STEP -4 :: CALL LOCATE(#1,43,I):: CALL SOUND(-10,110,0,600,4
,300,3,-5,0):: NEXT I
```

```
690 CALL CHAR(140,SC$(3)):: FOR I=43 TO 97 STEP 3 :: CALL LOCATE(#1,I,81):: CALL
 SOUND(-10,1000,0,900,0,880,0,-4,0):: NEXT I
700 CALL CHAR(140,SC$(2)):: FOR I=81 TO 153 STEP 4 :: CALL LOCATE(#1,97,I):: CAL
L SOUND(-10,110,0,600,4,300,3,-5,0):: NEXT I
710 CALL CHAR(140,SC$(3)):: FOR I=97 TO 153 STEP 4 :: CALL LOCATE(#1,I,153):: CA
LL SOUND(-10,1000,0,900,0,880,0,-4,0):: NEXT I
720 CALL CHAR(140,SC$(2)):: FOR I=153 TO 237 STEP 4 :: CALL LOCATE(#1,153,I):: C
ALL SOUND(-10,110,0,600,4,300,3,-5,0):: NEXT I
730 CALL CHAR(140,SC$(1)):: RP=1 :: LI=100 :: GOSUB 1490
740 CALL HCHAR(8,11,116,16):: CALL VCHAR(9,11,118,6):: CALL VCHAR(9,25,118,6)::
CALL HCHAR(10,6,126,23)
750 CALL HCHAR(13,17,118,13):: CALL HCHAR(15,5,121):: CALL HCHAR(15,20,120):: CA
LL VCHAR(16,5,118,6):: CALL VCHAR(16,20,118,6)
760 CALL HCHAR(17,5,126,23):: CALL HCHAR(20,12,118,13)
770 CALL DELSPRITE(#2,#3):: RANDOMIZE
780 IF R1+R2=20 THEN IF RND*3>2 THEN LS(1)=Z :: CALL VCHAR(8,L(1,LS(1)),117,7)
790 IF RND*2>.9 THEN LS(2)=Z :: CALL VCHAR(15,L(2,LS(2)),117,7)
800 FOR I=1 TO 2 :: IF RND*3>1.9 THEN IF RND*2<1 THEN CALL HCHAR(3+I*7,RZ(I,Z),1
27):: GOTO 820 ELSE 810 ELSE 820
810 IF (R1=10 AND I=1)OR(R2=10 AND I=2)THEN 820 ELSE SV=Z :: CALL HCHAR(3+I*7,RZ
(I,SV),123):: CALL HCHAR(3+I*7,RZ(I,SV)+1,116):: CALL HCHAR(3+I*7,RZ(I,SV)+2,122
)
820 IF RND*2<1 THEN IF Z=1 THEN CALL HCHAR(6+I*7,RH(I,Z),119):: GOTO 840 ELSE 83
0 ELSE 840
830 SV=Z :: CALL HCHAR(6+I*7,RH(I,SV),125):: CALL HCHAR(6+I*7,RH(I,SV)+1,124)
840 SPC=INT(RND*3+1):: CALL SPRITE(#I+1,124+SPC*4,SF(SPC),43+I*56,50+(Z-1)*100,0
,SP(SPC))
850 NEXT I
860 GOSUB 1080
870 CALL COINC(ALL,FG):: IF FG THEN 1330 ELSE CALL JOYST(1,T,Y):: CALL KEY(1,KT,
KY):: CALL MOTION(#1,0,T)
880 IF T<>0 THEN RP=1.5+SGN(T)*.5 :: CALL CHAR(140,SC$(RP)):: CALL SOUND(-200,11
0,0,600,4,800,3,-5,0)
890 IF KT=18 THEN 1180
900 CALL POSITION(#1,SZ,SS):: IF SS<16 OR SS>240 THEN 1300
910 IF Y=-4 THEN 990 ELSE IF Y=4 THEN 1090
920 CALL COINC(ALL,FG):: IF FG THEN 1330
930 CALL GCHAR(13+7*SGN(SZ-97),INT((SS+5+RP)/8),GFG):: CALL GCHAR(13+7*SGN(SZ-97
),INT((SS+13+RP)/8),GGF):: IF GGF+GFG>236 THEN 950
940 Q=Q+1 :: IF Q>=25 THEN 860 ELSE 870
950 CALL MOTION(#1,0,0):: CALL HCHAR(13+7*SGN(SZ-97),INT((SS+5+RP)/8)-1,118,4)
960 IF GGF+GFG>240 THEN RO=RO+1 ELSE HA=HA+1
970 CALL SOUND(-300,880,0,800,0,-4,0)
980 PU=PU+20*Z :: GOSUB 1470 :: Q=Q+10 :: GOTO 940
990 ! NACH UNTEN
1000 IF SZ=153 THEN 940
1010 CALL GCHAR(15,INT((SS+5+RP)/8),GFG):: CALL GCHAR(15,INT((SS+13+RP)/8),GGF)
:: IF GGF+GFG>239 THEN 940
1020 CALL MOTION(#1,0,0):: CALL CHAR(140,SC$(3)):: CALL LOCATE(#1,115,LS(2)*120-
87):: CALL MOTION(#1,-Y,0)
1030 CALL COINC(ALL,FG):: IF FG THEN 1330 ELSE CALL JOYST(1,T,Y):: IF Y<>-4 THEN
 Y=0
1040 IF Y=-4 THEN CALL SOUND(-200,1000,0,900,0,880,0,-4,0)
1050 CALL MOTION(#1,-Y,0):: CALL POSITION(#1,SZ,SS):: IF SZ>153 THEN SZ=153 ELSE
 1070
1060 CALL MOTION(#1,0,0):: CALL LOCATE(#1,SZ,SS):: RP=1 :: CALL CHAR(140,SC$(1))
:: Q=Q+10 :: GOTO 940
1070 Q=Q+5 :: IF Q>=25 THEN GOSUB 1080 :: GOTO 1030 ELSE 1030
1080 Q=Q-25 :: LI=LI-2.5 :: GOSUB 1490 :: IF LI<=0 THEN 1330 ELSE RETURN
1090 ! NACH OBEN
1100 IF SZ=97 AND R1+R2<20 THEN 940
1110 CALL GCHAR(13+7*SGN(SZ-97),INT((SS+5+RP)/8),GFG):: CALL GCHAR(13+7*SGN(SZ-9
7),INT((SS+13+RP)/8),GGF):: IF GGF+GFG>235 THEN 940
1120 CALL MOTION(#1,0,0):: CALL CHAR(140,SC$(3)):: CALL LOCATE(#1,SZ,LS(SGN(SZ-9
7)+1)*120-39-SGN(SZ-97)*48):: CALL MOTION(#1,-Y,0)
1130 CALL COINC(ALL,FG):: IF FG THEN 1330 ELSE CALL JOYST(1,T,Y):: IF Y<>4 THEN
Y=0
1140 IF Y=4 THEN CALL SOUND(-200,1000,0,900,0,880,0,-4,0)
1150 CALL MOTION(#1,-Y,0):: CALL POSITION(#1,SZ,SS):: IF SZ<115 AND SZ>97 THEN 1
160 ELSE IF SZ<57 THEN 1440 ELSE 1170
1160 CALL MOTION(#1,0,0):: SZ=97 :: CALL LOCATE(#1,SZ,SS):: RP=1 :: CALL CHAR(14
0,SC$(1)):: Q=Q+10 :: GOTO 940
1170 Q=Q+5 :: IF Q>=25 THEN GOSUB 1080 :: GOTO 1130 ELSE 1130
1180 CALL MOTION(#1,0,0):: CALL POSITION(#1,SZ,SS)
```

```
1190 IF (RP=1 AND SS<44)OR(RP=2 AND SS>212)THEN 940
1200 CALL MOTION(#1,0,0):: FOR I=1 TO 12 :: SZ=SZ-2 :: SS=SS+SGN(RP-1.5):: CALL
LOCATE(#1,SZ,SS):: CALL SOUND(-200,400+I*20,1):: NEXT I
1210 Q=0 :: LI=LI-5 :: IF LI<0 THEN LI=0
1220 GOSUB 1490
1230 CALL GCHAR(10+7*SGN(SZ-73),INT((SS+5+RP)/8),GFG):: CALL GCHAR(10+7*SGN(SZ-7
3),INT((SS+13+RP)/8),GGF):: IF GGF+GFG=252 THEN 1280
1240 IF GGF=117 OR GFG=117 THEN 1290 ELSE IF GFG+GGF=253 THEN LI=100 :: GOSUB 14
90 :: PU=PU+50 :: GOTO 1270
1250 IF HA=0 OR RO=0 THEN 1290 ELSE IF SZ=73 THEN R1=R1+1 ELSE R2=R2+1
1260 RO=RO-1 :: HA=HA-1 :: PU=PU+70+20*Z
1270 CALL SOUND(-200,1000,0,110,3,330,2,-3,0):: GOSUB 1470 :: CALL HCHAR(10+7*SG
N(SZ-73),INT((SS+5+RP)/8)-2,126,6)
1280 IF LI<=0 THEN 1330
1290 FOR I=1 TO 12 :: SZ=SZ+2 :: SS=SS+SGN(RP-1.5):: CALL LOCATE(#1,SZ,SS):: CAL
L SOUND(-200,660-I*20,1):: NEXT I :: Q=Q+5 :: GOTO 940
1300 !BILDWECHSEL
1310 CALL MOTION(#1,0,0):: IF RP=1 THEN CALL LOCATE(#1,SZ,236)ELSE CALL LOCATE(#
1,SZ,16)
1320 GOTO 740
1330 !ENDE
1340 CALL MOTION(#1,0,0):: CALL HCHAR(1,28-AS,32):: AS=AS-1
1350 DATA 2,659,523,440,2,523,523,523,2,659,494,392,2,494,494,494,2,587,440,349
1360 DATA 2,523,523,523,1,494,392,494,2,523,523,523,2,587,587,587,1,523,392,659
1370 RESTORE 1350 :: FOR I=1 TO 10 :: READ D,F1,F2,F3 :: CALL SOUND(600/D,F1,0,F
2,0,F3,0):: NEXT I
1380 RESTORE 1360 :: FOR I=1 TO 5 :: READ D,F1,F2,F3 :: CALL SOUND(600/D,F1,0,F2
,0,F3,0):: NEXT I
1390 IF AS>0 THEN 660
1400 M=-500 :: FOR II=1 TO 5 :: RESTORE 230 :: M=M+210
1410 FOR I=1 TO 13 :: DISPLAY AT(10,6)SIZE(16):"G A M E  O V E R" :: READ F1 ::
CALL SOUND(M,F1,0,F1*2,0,F1*3,0):: CALL KEY(1,KT,KY):: IF KT=18 THEN 1430
1420 DISPLAY AT(10,6)SIZE(16):"                " :: NEXT I :: NEXT II :: GOTO 14
00
1430 CALL DELSPRITE(ALL):: GOTO 470
1440 ! GESCHAFFT
1450 CALL MOTION(#1,0,0):: DISPLAY AT(18,1):"  DU HAST ES GESCHAFFT UND     WIRS
T ZUM KANALMEISTER         BEFOERDERT"
1460 GOSUB 190 :: CALL DELSPRITE(ALL):: GOTO 470
1470 DISPLAY AT(23,7)SIZE(3):R1 :: DISPLAY AT(23,17)SIZE(3):R2
1480 DISPLAY AT(23,26)SIZE(3):RO :: DISPLAY AT(24,7)SIZE(3):HA :: DISPLAY AT(1,1
1)SIZE(6):PU :: RETURN
1490 IF LI=100 THEN CALL HCHAR(24,18,112,10):: GOTO 1540
1500 FOR I=0 TO 3 :: IF INT(LI/10)=LI/10-I*.25 THEN LC=112+I :: GOTO 1520
1510 NEXT I
1520 IF LI=0 THEN 1530 ELSE CALL HCHAR(24,18+INT((LI-1)/10),LC)
1530 IF LI<90 THEN CALL HCHAR(24,18+LI/10,116)
1540 RETURN
1550 PRINT "SPIELERLAEUTERUNG":;:"DU MUSST 20 LECKS IN EINER"
1560 PRINT :"WASSERLEITUNG REPARIEREN !":;:"FUER JEDE REPARATUR "
1570 PRINT :"BRAUCHST DU EINEN HAMMER UND":;:"EIN ROHRSTUECK,DIESE LIEGEN"
1580 PRINT :"IN DEN GAENGEN. IN JEDER":;:"ETAGE SIND 10 LECKS. HAST DU"
1590 PRINT :"ALLE 20 REPARIERT KANNST DU":;:"WIEDER AN DIE OBERFLAECHE."
1600 PRINT :TAB(26);">>"
1610 CALL KEY(1,KY,KT):: IF KT=0 THEN 1610 ELSE CALL CLEAR
1620 PRINT "G!ACHTUNG!ACHTUNG!ACHTUNG!AC":;:"DU KANNST NUR SOLANGE IN"
1630 PRINT :"DEN GAENGEN LAUFEN,SOLANGE":;:"NOCH LICHT GENUG SCHEINT"
1640 PRINT :"(ANZEIGE RECHTS UNTEN). UM":;:"DEN LICHTVORRAT WIEDER AUF"
1650 PRINT :"ZU FUELLEN MUSST DU AN DEN":;:"LICHTKASTEN ";CHR$(127);" ANSPRINGEN
."
1660 PRINT :"AUCH DIE LECKS ";CHR$(123);CHR$(116);CHR$(122);" MUSST":;:"DU ANSPR
INGEN UM SIE ZU RE-"
1670 PRINT :"PARIEREN.                    >>"
1680 CALL KEY(1,KT,KY):: IF KY=0 THEN 1680 ELSE CALL CLEAR
1690 PRINT "UEBER DIE GIFTWOLKEN,WELLEN":;:"UND KROKODILE, DIE AUF DEN"
1700 PRINT :"ABWAESSERN SCHWIMMEN, MUSST":;:"DU SPRINGEN. AUF DEN LEITERN"
1710 PRINT :"KANNST DU DICH NUR IN DIE":;:"RICHTUNG BEWEGEN,IN DER"
1720 PRINT :"DU SIE BETRETEN HAST.      >>"
1730 CALL KEY(1,KT,KY):: IF KY=0 THEN 1730 ELSE CALL CLEAR
1740 PRINT "STEUERUNG :"
1750 PRINT :"JOYSTICK 1 + FEUERKNOPF":
1760 PRINT :"'ALPHA-LOCK' BITTE LOESEN !"
1770 PRINT :TAB(26);">>"
1780 CALL KEY(1,KT,KY):: IF KY=0 THEN 1780 ELSE CALL CLEAR :: EI=1 :: GOTO 400
```

SANTA MARIA

TI-99/4A

mit Extended Basic-Modul

Bergung des Flaggschiffs „Santa Maria“

Sie haben einen alten Plan in die Hände bekommen, in dem die ungefähre Lage eines alten Schiffes, der „Santa Maria“, verzeichnet ist. Dieses Schiff wollen Sie nun heben.

Dazu müssen Sie

1. das Wrack finden,
2. so viele Luftbehälter im Wrack unterbringen, wie nötig sind, um es an die Wasseroberfläche zu bringen,
3. das Wrack in den Hafen schleppen.

Für die Suche steht Ihnen ein Echolot zur Verfügung. Sie bewegen Ihr Schiff in kleinen Schritten mit den Pfeiltasten „S“ nach links und „D“ nach rechts. Mit „X“ wird es gestoppt. Jetzt wird mit „E“ das Echolot betätigt und auf das Ergebnis gewartet. Nachdem das Echo zu hören war, kann das Schiff weiter bewegt werden. Haben Sie die Position des Wracks geortet, wird dies durch einen anderen Ton des Echos angezeigt.

Jetzt wird mit der Taste „T“ der Taucher ins Wasser gebracht. Um dem Wrack genügend Auftrieb zu verleihen, müssen von den 40 Luftbehältern, die Sie zur Verfügung haben, 10 Behälter im Wrack untergebracht werden.

Aber Vorsicht! Auf dem Weg zum Wrack werden Sie durch viele Fische behindert, die nur darauf warten, daß Sie ihnen zu nahe kommen, denn dann beißen sie Ihnen den Luftbehälter kaputt. In diesem Fall bleibt Ihnen nichts anderes übrig, als wieder aufzutauchen, um einen neuen Luftbehälter zu holen.

Sind Sie weit genug getaucht, wechselt das Bild und Sie sehen das Wrack unter sich auf dem Meeresboden liegen. Ist das Wrack erreicht, gilt der Behälter als untergebracht und Sie werden mit einem Punkt belohnt. Tauchen Sie jetzt mit der Taste „E“ wieder auf und holen Sie sich den nächsten Luftbehälter. Wenn Sie 10 Behälter untergebracht haben, taucht das Wrack auf.

Nun können Sie es mit der Taste „D“ in den Hafen schleppen. Da sich das Wetter während der Bergungsaktion zusehends verschlechtert hat, kann es Ihnen auf dem Weg in den schützenden Hafen passieren, daß Sie in ein Gewitter geraten und das Wrack, von einem Blitz getroffen, wieder im Meer versinkt.

```
100 !*** SANTA MARIA ***
110 !
120 CALL TITEL
130 CALL CLEAR :: CALL SCREEN(2):: CALL MAGNIFY(4):: RANDOMIZE
140 HIMMEL$=RPT$("55AA",4)
150 WOLKE1$="0000000000000001F7F7FFFFFFFFF7D3800000E1F3F7FFFFFFFFFFFFFFFFFFFEFC"
160 WOLKE2$="1E3F7FFFFFFF7F3F7FFFFFFFFF7F3F1F00008080809EBF7FFFFFFFFEFCF8E000"
170 WASSER$=RPT$("F",16)
180 BOOT$="0000000000F0080406091013121F1F0F0000000000000000000080F848FFFEFC"
190 ECHO$=RPT$("8000",4)
200 FISCH$=RPT$("0",19)&"61C2EFFFF3E1C"&RPT$("0",20)&"60C08080C060"
210 KRAKE$=RPT$("0",46)&"3C5A7E3C5AA5A5A581"
220 TAUCHER$="0000000000000000101000305050102020000000000000C0C080E0D0D0C02020"
230 BEHAELTER$=RPT$("0",50)&RPT$("0F",6)
240 GRUND$=RPT$("F",16)
250 WRACK1$=RPT$("0",53)&"1F8AEFF7F7F"
260 WRACK2$=RPT$("0",14)&"80A1C1838581FFFFFF"&RPT$("0",16)&"4080000007FEFCF8"
270 BLASEN$="8009200440022008"
280 HF1,HF2=8 :: WF1=16 :: WF2=15 :: WZ=144
290 P,Z,ZE=0 :: LB=40
300 CALL CHAR(40,HIMMEL$,33,WASSER$,96,WOLKE1$,100,WOLKE2$,104,WOLKE1$,108,WOLKE
2$,112,BOOT$,120,ECHO$)
310 CALL CHAR(124,FISCH$,128,KRAKE$,132,TAUCHER$,60,BEHAELTER$,122,GRUND$,136,WR
ACK1$,140,WRACK2$,123,BLASEN$)
320 CALL COLOR(2,HF1,HF2,1,5,5,12,12,5)
330 FOR I=3 TO 8 :: CALL COLOR(I,8,5):: NEXT I
340 CALL HCHAR(1,1,40,256):: CALL HCHAR(9,1,33,512)
350 DISPLAY AT(24,2)SIZE(10):"BEHAELTER:" :: DISPLAY AT(24,12)SIZE(3):LB :: DISP
LAY AT(24,19)SIZE(7):"PUNKTE:" :: DISPLAY AT(24,26):P
360 !
370 !*** WOLKEN ***
380 CALL SPRITE(#2,96,WF1,17,40,#3,100,WF1,17,68,#4,104,WF2,1,120,#5,108,WF2,1,1
48)
390 CALL MOTION(#2,0,2,#3,0,2,#4,0,1,#5,0,1)
400 !
410 !*** FISCHE ***
420 RANDOMIZE :: CALL SPRITE(#6,124,7,99,100,0,-(INT(RND*20)+6)):: RANDOMIZE ::
CALL SPRITE(#7,124,4,147,150,0,-(INT(RND*30)+6))
430 RANDOMIZE :: CALL SPRITE(#8,128,11,120,40,0,INT(RND*20)+6):: RANDOMIZE :: CA
LL SPRITE(#9,128,14,74,110,0,INT(RND*30)+6)
440 !
450 !*** WRACKPOSITION ***
```

```
460 WP=INT((205-41+1)*RND)+41
470 !
480 !*** BOOT ***
490 CALL SPRITE(#1,112,2,33,1)
500 CALL KEY(0,K,S):: IF S=0 THEN 500
510 IF K=69 THEN CALL ECHOLOT(E,WP,SZ,SS)
520 CALL WETTER(E,LB,HF1,HF2)
530 IF K=68 THEN CALL MOTION(#1,0,2)ELSE IF K=83 THEN CALL MOTION(#1,0,-2)ELSE I
F K=88 THEN CALL MOTION(#1,0,0)
540 IF SS<(WP-3)OR SS>(WP+3)THEN 500
550 !
560 TZ=SZ+14
570 CALL KEY(0,K,S):: IF S=0 OR K<>84 THEN 570
580 IF K=84 THEN CALL SPRITE(#10,132,2,TZ,SS-16,#11,60,15,TZ,SS-20):: LB=LB-1 ::
 DISPLAY AT(24,12)SIZE(3):LB
590 CALL COINC(#11,#6,10,J):: IF J=-1 THEN 690
600 CALL COINC(#11,#7,10,J):: IF J=-1 THEN 690
610 CALL COINC(#11,#8,10,J):: IF J=-1 THEN 690
620 CALL COINC(#11,#9,10,J):: IF J=-1 THEN 690
630 CALL KEY(0,K,S):: IF S=0 THEN 590
640 IF TZ>169 THEN 850
650 IF K=88 AND TZ<=169 THEN CALL SPRITE(#10,132,2,TZ+4,SS-16,#11,60,15,TZ+4,SS-
20):: TZ=TZ+4
660 IF K=69 AND TZ>=SZ+14 THEN CALL SPRITE(#10,132,2,TZ-2,SS-16,#11,60,15,TZ-2,S
S-20):: TZ=TZ-2
670 CALL WETTER(E,LB,HF1,HF2)
680 GOTO 590
690 CALL DELSPRITE(#11):: CALL SOUND(400,-6,2)
700 FOR I=INT((TZ+8)/8)TO INT((SZ+14)/8)+4 STEP -1
710 CALL HCHAR(I,INT((SS-10)/8)+3,123):: CALL HCHAR(I+1,INT((SS-10)/8)+3,121)::
CALL HCHAR(9,1,121;32)
720 NEXT I
730 CALL KEY(0,K,S):: IF S=0 THEN 730
740 IF K<>69 THEN 730
750 IF K=69 AND TZ>=SZ+14 THEN CALL SPRITE(#10,132,2,TZ-2,SS-16):: TZ=TZ-2
760 IF TZ>=SZ+14 THEN 730
770 IF LB>0 THEN CALL SPRITE(#10,132,2,TZ,SS-16,#11,60,15,TZ,SS-20):: LB=LB-1 ::
 DISPLAY AT(24,12)SIZE(3):LB :: GOTO 630
780 CALL WETTER(E,LB,HF1,HF2)
790 IF LB=0 THEN DISPLAY AT(14,1):"SIE HABEN ES NICHT GESCHAFFT"
800 DISPLAY AT(16,1):"VON 40 LUFTBEHAELTERN        KONNTEN SIE NUR";P;"IM WRACK U
NTERBRINGEN"
810 DISPLAY AT(20,1):"NOCH EIN VERSUCH  J ODER N"
820 GOTO 1520
830 !
840 !*** BILD 2 ***
850 TZ=4
860 FOR I=1 TO 5 :: CALL DELSPRITE(#I):: NEXT I
870 CALL DELSPRITE(#10,#11)
880 CALL HCHAR(1,1,33,672):: CALL HCHAR(22,1,122,96)
890 DISPLAY AT(1,2)SIZE(10):"BEHAELTER:" :: DISPLAY AT(1,12)SIZE(3):LB :: DISPLA
Y AT(1,19)SIZE(7):"PUNKTE:" :: DISPLAY AT(1,26):P
900 CALL SPRITE(#12,136,2,144,WP-36,#13,140,2,144,WP-4)
910 CALL SPRITE(#10,132,2,TZ,SS-16,#11,60,15,TZ,SS-20)
920 RANDOMIZE :: CALL SPRITE(#6,124,7,25,150,0,-(INT(RND*30)+6)):: RANDOMIZE ::
CALL SPRITE(#7,124,4,120,100,0,-(INT(RND*30)+6))
930 RANDOMIZE :: CALL SPRITE(#8,128,11,90,40,0,INT(RND*30)+6):: RANDOMIZE :: CAL
L SPRITE(#9,128,14,55,110,0,INT(RND*30)+6)
940 CALL COINC(#11,#6,10,J):: IF J=-1 THEN 1040
950 CALL COINC(#11,#7,10,J):: IF J=-1 THEN 1040
960 CALL COINC(#11,#8,10,J):: IF J=-1 THEN 1040
970 CALL COINC(#11,#9,10,J):: IF J=-1 THEN 1040
980 CALL KEY(0,K,S):: IF S=0 THEN 940
990 IF TZ<3 THEN 1170
1000 IF TZ>=144 THEN P=P+1 :: DISPLAY AT(1,26):P :: GOSUB 1560 :: CALL DELSPRITE
(#11):: GOTO 1050
1010 IF K=88 AND TZ<=150 THEN CALL SPRITE(#10,132,2,TZ+4,SS-16,#11,60,15,TZ+4,SS
-20):: TZ=TZ+4
1020 IF K=69 AND TZ>=3 THEN CALL SPRITE(#10,132,2,TZ-2,SS-16,#11,60,15,TZ-2,SS-2
0):: TZ=TZ-2
1030 GOTO 940
1040 CALL DELSPRITE(#11):: CALL SOUND(400,-6,2)
1050 IF P=10 THEN CALL STEIGEN(WZ,WP,SZ,SS,WF1,WF2,P,LB)
1060 IF WZ<=SZ THEN 1290
1070 IF TZ>=144 THEN 1120
```

```
1080 FOR I=INT((TZ+8)/8)TO 1 STEP -1
1090 CALL HCHAR(I,INT((SS-10)/8)+3,123):: CALL HCHAR(I+1,INT((SS-10)/8)+3,121)::
 NEXT I
1100 CALL HCHAR(1,1,121,32):: DISPLAY AT(1,2)SIZE(10):"BEHAELTER:" :: DISPLAY AT
(1,12)SIZE(3):LB
1110 DISPLAY AT(1,19)SIZE(7):"PUNKTE:" :: DISPLAY AT(1,26):P
1120 CALL KEY(0,K,S):: IF S=0 THEN 1120
1130 IF K<>69 THEN 1120
1140 IF K=69 AND TZ>=3 THEN CALL SPRITE(#10,132,2,TZ-2,SS-16):: TZ=TZ-2
1150 IF TZ>=3 THEN 1120
1160 !
1170 !*** BILD 1 ***
1180 TZ=169
1190 FOR I=6 TO 13 :: CALL DELSPRITE(#I):: NEXT I
1200 CALL HCHAR(22,1,33,96):: CALL HCHAR(1,1,40,256)
1210 CALL SPRITE(#1,112,2,33,SS)
1220 CALL SPRITE(#2,96,WF1,17,40,#3,100,WF1,17,68,#4,104,WF2,1,120,#5,108,WF2,1,
148)
1230 CALL MOTION(#2,0,2,#3,0,2,#4,0,1,#5,0,1)
1240 CALL SPRITE(#10,132,2,TZ,SS-16)
1250 RANDOMIZE :: CALL SPRITE(#6,124,7,99,100,0,-(INT(RND*20)+6)):: RANDOMIZE ::
 CALL SPRITE(#7,124,4,147,150,0,-(INT(RND*30)+6))
1260 RANDOMIZE :: CALL SPRITE(#8,128,11,120,40,0,INT(RND*20)+6):: RANDOMIZE :: C
ALL SPRITE(#9,128,14,74,110,0,INT(RND*30)+6)
1270 DISPLAY AT(24,2)SIZE(10):"BEHAELTER:" :: DISPLAY AT(24,12)SIZE(3):LB :: DIS
PLAY AT(24,19)SIZE(7):"PUNKTE:" :: DISPLAY AT(24,26):P
1280 GOTO 730
1290 CALL KEY(0,K,S):: IF S=0 OR K<>68 THEN 1290
1300 IF K=68 THEN CALL MOTION(#1,0,2,#12,0,2,#13,0,2)
1310 CALL POSITION(#1,ZE,SP)
1320 IF SP>=206 THEN 1360
1330 GOTO 1310
1340 !
1350 !*** BILD 3 ***
1360 FOR I=6 TO 9 :: CALL DELSPRITE(#I):: NEXT I
1370 CALL CHAR(123,RPT$("F",16),124,RPT$("0",16),119,RPT$("0",8)&RPT$("F",8),128
,RPT$("F",16))
1380 CALL COLOR(13,13,13,12,9,16,11,2,15)
1390 CALL HCHAR(8,20,128,13)
1400 FOR I=7 TO 4 STEP -2 :: CALL HCHAR(I,21,123,3):: CALL HCHAR(I-1,21,124,3)::
 NEXT I
1410 CALL HCHAR(3,20,123,5):: CALL HCHAR(2,21,124,3):: CALL HCHAR(1,22,119)
1420 CALL LOCATE(#12,WZ,2,#13,WZ,33,#1,33,60)
1430 CALL GEWITTER(LB,Z,S,BP)
1440 IF Z>=160 THEN 1500
1450 CALL POSITION(#1,ZE,SP)
1460 IF SP>=215 THEN CALL MOTION(#1,0,0,#12,0,0,#13,0,0):: GOTO 1480
1470 GOTO 1450
1480 CALL HCHAR(24,1,33,32)
1490 DISPLAY AT(14,1):"HERZLICHEN GLUECKWUNSCH      SIE HABEN ES GESCHAFFT"
1500 DISPLAY AT(17,1):"VIELLEICHT SIND SIE BEI DER NAECHSTEN EXPEDITION WIEDER D
ABEI"
1510 DISPLAY AT(21,1):"WAEHLEN SIE J ODER N"
1520 CALL KEY(0,K,S):: IF S=0 THEN 1520
1530 IF K<>74 AND K<>78 THEN 1520
1540 IF K=74 THEN CALL CLEAR :: CALL DELSPRITE(ALL):: CALL CHARSET :: GOTO 130
1550 CALL DELSPRITE(ALL):: CALL CLEAR :: END
1560 FOR I=1 TO 10 :: CALL SOUND(-100,220*I,0):: NEXT I
1570 RETURN
1580 !********************
1590 !
1600 SUB ECHOLOT(E,WP,SZ,SS)
1610 E=E+1
1620 CALL POSITION(#1,SZ,SS)
1630 IF SS<1 OR SS>=240 THEN SUBEXIT
1640 CALL CHAR(121,"")
1650 FOR I=10 TO 23 :: CALL HCHAR(I,SS/8+3,120):: CALL HCHAR(I-1,SS/8+3,121):: N
EXT I :: CALL HCHAR(23,SS/8+3,121)
1660 FOR I=1 TO 500 :: NEXT I
1670 FOR I=0 TO 10 STEP 2
1680 IF SS<(WP-3)OR SS>(WP+3)THEN CALL SOUND(100,1760,I)ELSE CALL SOUND(100,262,
I)
1690 NEXT I
1700 SUBEND
```

```
1710 !
1720 !
1730 SUB STEIGEN(WZ,WP,SZ,SS,WF1,WF2,P,LB)
1740 CALL DELSPRITE(#10)
1750 CALL MOTION(#12,-2,0,#13,-2,0)
1760 CALL POSITION(#12,WZ,WS)
1770 IF WZ<=3 THEN 1790
1780 GOTO 1760
1790 WZ=169
1800 FOR I=6 TO 13 :: CALL DELSPRITE(#I):: NEXT I
1810 CALL HCHAR(22,1,33,96):: CALL HCHAR(1,1,40,256)
1820 CALL SPRITE(#1,112,2,33,SS+20)
1830 CALL SPRITE(#2,96,WF1,17,40,#3,100,WF1,17,68,#4,104,WF2,1,120,#5,108,WF2,1,
148)
1840 CALL MOTION(#2,0,2,#3,0,2,#4,0,1,#5,0,1)
1850 CALL SPRITE(#12,136,2,WZ,WP-40,#13,140,2,WZ,WP-8)
1860 RANDOMIZE :: CALL SPRITE(#6,124,7,99,100,0,-(INT(RND*20)+6)):: RANDOMIZE ::
 CALL SPRITE(#7,124,4,147,150,0,-(INT(RND*30)+6))
1870 RANDOMIZE :: CALL SPRITE(#8,128,11,120,40,0,INT(RND*20)+6):: RANDOMIZE :: C
ALL SPRITE(#9,128,14,74,110,0,INT(RND*30)+6)
1880 DISPLAY AT(24,2)SIZE(10):"BEHAELTER:" :: DISPLAY AT(24,12)SIZE(3):LB :: DIS
PLAY AT(24,19)SIZE(7):"PUNKTE:" :: DISPLAY AT(24,26):P
1890 CALL MOTION(#12,-2,0,#13,-2,0)
1900 CALL POSITION(#12,WZ,WS)
1910 IF WZ<=SZ THEN CALL MOTION(#12,0,0,#13,0,0):: GOTO 1930
1920 GOTO 1900
1930 SUBEND
1940 !
1950 SUB WETTER(E,LB,HF1,HF2)
1960 IF LB=30 OR E=30 THEN HF2=15 :: GOTO 2000
1970 IF LB=20 THEN HF1,HF2=15 :: GOTO 2000
1980 IF LB=10 THEN HF1=8 :: GOTO 2000
1990 IF LB=5 THEN HF2=11
2000 IF LB=30 OR E=30 OR LB=20 OR LB=10 OR LB=5 THEN CALL COLOR(2,HF1,HF2):: CAL
L HCHAR(1,1,40,256)
2010 SUBEND
2020 !
2030 SUB GEWITTER(LB,Z,S,BP)
2040 IF LB<=5 THEN BP=INT((120-10+1)*RND)+10 ELSE SUBEXIT
2050 CALL POSITION(#12,Z,S)
2060 IF S>=BP THEN 2080
2070 GOTO 2050
2080 CALL CHAR(132,"00001008040602020301"&RPT$("0",24)&"80C040202020201008"):: C
ALL PATTERN(#10,132):: CALL DELSPRITE(#4,#5)
2090 FOR I=1 TO 2 :: CALL SPRITE(#10,132,16,5,30):: FOR J=1 TO 50 :: NEXT J :: C
ALL DELSPRITE(#10):: NEXT I
2100 CALL SOUND(100,-5,0):: CALL SOUND(400,-5,0)
2110 CALL MOTION(#12,2,0,#13,2,0)
2120 CALL HCHAR(24,1,33,32):: DISPLAY AT(13,1):"LEIDER HAT IHNEN DAS WETTER EINE
N STRICH DURCH DIE      RECHNUNG GEMACHT"
2130 CALL MOTION(#1,0,0)
2140 CALL POSITION(#12,Z,S)
2150 IF Z>=160 THEN CALL DELSPRITE(#12,#13):: GOTO 2170
2160 GOTO 2140
2170 SUBEND
2180 !
2190 SUB TITEL
2200 CALL CLEAR :: CALL SCREEN(5):: CALL MAGNIFY(2)
2210 DATA 83,65,78,84,65,32,77,65,82,73,65
2220 CALL CHAR(96,"8009200440022008",104,"")
2230 CALL COLOR(9,11,5,10,5,5)
2240 T1$="00061C2EFFFF3E1C"
2250 T2$="000060C08080C060"
2260 T3$=RPT$("F",16)
2270 CALL CHAR(124,T1$,125,T2$,126,T3$)
2280 CALL SPRITE(#1,126,5,120,1,#4,124,11,120,150,#5,125,11,120,165,#2,126,5,180
,45,#3,126,5,180,53)
2290 CALL MOTION(#4,0,-15,#5,0,-15)
2300 W=40 :: X=55 :: Y=X-10 :: Z=6
2310 FOR I=1 TO 11
2320 CALL POSITION(#4,ZE,SP)
2330 IF SP<X AND SP>Y THEN CALL MOTION(#4,0,0,#5,0,0):: GOTO 2350
2340 GOTO 2320
2350 READ T
```

```
2360 FOR K=INT(ZE/8)TO (W/8)+1 STEP -1
2370 CALL VCHAR(ZE/8,SP/8,96)
2380 CALL VCHAR(INT(ZE/8)+1,SP/8,104):: CALL VCHAR(INT(ZE/8),SP/8,104)
2390 ZE=ZE-8
2400 NEXT K
2410 CALL SPRITE(#Z,T,11,W,X-10):: W=W+16 :: X=X+16 :: Y=Y+16 :: Z=Z+1
2420 IF W=88 THEN W=40
2430 FOR J=1 TO 50 :: NEXT J :: CALL MOTION(#4,0,-15,#5,0,-15)
2440 NEXT I
2450 CALL CHAR(64,"3C4299A1A199423C")
2460 Y$=CHR$(64)&" 1984  BY WOLFGANG LENK"
2470 DISPLAY AT(19,3):Y$
2480 DISPLAY AT(22,3):"BITTE EINE TASTE DRUECKEN"
2490 CALL KEY(0,K,S):: IF S=0 THEN 2490
2500 FOR I=6 TO 16 :: CALL MOTION(#I,-5,0):: FOR J=1 TO 30 :: NEXT J :: NEXT I
2510 CALL POSITION(#8,ZEI,SPA)
2520 IF ZEI=1 THEN 2540
2530 GOTO 2510
2540 FOR I=6 TO 12 STEP 3 :: FOR J=1 TO 10 :: NEXT J :: CALL DELSPRITE(#I,#I+3,#
I+6):: NEXT I
2550 CALL DELSPRITE(ALL):: CALL CHARSET
2560 SUBEND
```

Korrektur „Infra" für TI-99/4A, Heft 4

Beachten Sie bitte nur die Zeile 650!

```
640 DISPLAY AT(23,1):"REPA1:     REPA2:     ROHR:  "
650 DISPLAY AT(24,1):"HAMM:      LICHTPPPPPPPPPP" :: DISPLAY AT(1,7)SIZE(4):"PUN:
" :: GOSUB 1470
660 CALL VCHAR(8,11,117,7):: CALL VCHAR(15,20,117,7):: CALL DELSPRITE(ALL)
```

TARZAN

TI-99/4A

TI-99/4A mit Extended Basic und Joystick

Der gefährliche Weg durch den Dschungel

In diesem Spiel müssen Sie als Tarzan drei gefährliche Abenteuer bestehen und dann Jane retten, die irgendwo im Dschungel gefangengehalten wird. Nach Erscheinen des Titelbildes wird das Spiel durch Drücken der Feuertaste gestartet. Im ersten Bild steht Tarzan auf einem Vorsprung und muß mit Hilfe von Lianen einen reißenden Fluß überqueren. Drücken Sie die Feuertaste, wenn Tarzan springen soll. Aber passen Sie auf! Tarzan kann nicht schneller springen, als die Liane schwingt! Er kann auch nicht auf den Vorsprung zurück. Das Abenteuer gilt als überstanden, wenn Tarzan nach einer bestimmten Anzahl von Lianen, die von Runde zu Runde größer wird, sicher auf dem linken Vorsprung landet.

Allerlei Untier muß Tarzan im zweiten Bild überspringen. Es handelt sich dabei um Steine, Affen und Pflanzen. Tarzan bewegt sich ohne Einwirkung des Spielers nach links. Zum Springen wird wieder die Feuertaste gedrückt. Auch hier gilt das Erreichen des linken Bildschirmrandes als Erfüllung der Aufgabe.

Jane ist im dritten Bild schon deutlich auf einem erhöhten Vorsprung links im Bild zu sehen. Tarzan muß aber noch zwei spinnenartige Wesen überwinden. Er darf keines dieser Tierchen berühren. Sobald er vor Jane auf der Säule steht, hat er Jane gerettet und die Aufgabe ist gelöst. Nun beginnt das Spiel von vorne, allerdings mit einem höheren Schwierigkeitsgrad. In diesem Bild ist Tarzan mit dem Joystick steuerbar (links, rechts, Feuer = Sprung).

```
A
100 ! *********TARZAN***
110 ! T                 *
120 ! A RALF MESCHEDE   T
130 ! R                 A
140 ! Z                 R
150 ! A                 Z
160 ! N                 A
170 ! *                 N
180 ! **TARZAN*********
190 !
```

Listing nächste Seite:

```
200 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL MAGNIFY(4)
210 DIM T$(2):: POY=0
220 T$(1)="000C1E2612120C1A2A2A2A2214080838"&RPT$("0",32)
230 T$(2)="00000102010108050203121 21F00000000C0E0602020C040A0A020408"
240 CALL CHAR(136,"30181830306060301818306030180C18"&RPT$("0",32)):: PU,R=0 :: M
F=3 :: CALL CHAR(140,T$(1))
250 CALL CHAR(132,RPT$("0",23)&"2070F1FF1"&RPT$("0",21)&"1ABFFFF06C3")
260 CALL CHAR(128,"0000000000000003060D9870E5FFBE008"&RPT$("0",33))
270 CALL CHAR(120,"FFFFFFFFFFFFFFFF",121,"FFFFFFFFFEFC702",122,"FFFFFFBF1F19101"
,123,"FFFFFFFFFFFEF464")
280 CALL CHAR(116,"0",117,"FFFFFFFFFFFFFFFF",118,"60F3FFFFFFFFFFFF",119,"0C3EFFF
FFFFFFFFF")
290 CALL CHAR(65,"040E8C0020481C3C",66,"3913033170783006",67,"C1431A3830120702",
68,"4072270210783")
300 CALL CHAR(72,"1F0F0F0783C1C0E"&RPT$("0",49))
310 CALL CHAR(76,"703010108F070381C0C0E0F008080C0E"&RPT$("0",32))
320 CALL CHAR(80,"000000000E0C0800E0C0800100000000E0C18103070F1F1F3070F0F0000000
00")
330 DATA 3C4299A1A199423C,00050D150405,00CC52CC52CC,00E785E726E5,000212725269,00
1A237222A2,00206EAE2826,0001DD9151DD
340 DATA 00001C9D514D,00405CDC508C
350 RESTORE 330 :: FOR I=48 TO 57 :: READ X$ :: CALL CHAR(I,X$):: NEXT I
360 CALL COLOR(11,5,4,12,13,4):: FOR I=0 TO 8 :: CALL COLOR(I,16,13):: NEXT I
370 CALL HCHAR(1,1,116,768)
380 CALL HCHAR(1,1,120,96):: DISPLAY AT(4,1):"(yz(yz(yz(yz(yz(yz(yz(yz(yz(" :: C
ALL HCHAR(4,1,121):: CALL HCHAR(4,2,122):: CALL HCHAR(4,31,121)
390 CALL HCHAR(4,32,122):: FOR I=1 TO 31 STEP 2 :: CALL HCHAR(20,I,118):: CALL H
CHAR(20,I+1,119):: NEXT I :: CALL HCHAR(21,1,117,128)
400 DISPLAY AT(2,7)SIZE(10):"0123456789" :: RANDOMIZE
410 CALL COLOR(5,2,11):: DISPLAY AT(6,1):"ABACDtDBCtBABtBACDtDBCtAttBtttBtttAtCt
AtCtttDCtAtCtDBtCtttDtttCABtBCDttABCtCABtCABDt"
420 DISPLAY AT(9,1):"ttCtttDtAtDABtBDAttDtAtBDCAtttDtttAtDtBCDtDCBAtAtDtCtADt"
430 CALL SPRITE(#4,72,4,65,105,0,0):: CALL SPRITE(#5,80,4,49,129,0,0):: CALL SPR
ITE(#3,76,4,49,209,0,0)
440 CALL SPRITE(#1,136,13,27,100):: CALL SPRITE(#2,136,13,59,100):: CALL MOTION(
#1,0,-10,#2,0,-10)
450 CALL SPRITE(#6,132,3,150,100):: CALL SPRITE(#7,128,3,150,132):: CALL MOTION(
#6,0,8,#7,0,8)
460 RESTORE 1670 :: FOR I=1 TO 14 :: READ ML,M1,M2,M3 :: CALL SOUND(400*ML,M1,2,
M2,0,M3,1):: XD=XD+9 :: IF XD<>11 THEN XD=2 :: DX=11 ELSE DX=2
470 CALL COLOR(5,XD,DX):: CALL KEY(1,T,Y):: IF Y THEN 520
480 NEXT I
490 FOR II=1 TO 2 :: RESTORE 1710 :: FOR I=1 TO 8 :: READ ML,M1,M2,M3 :: CALL SO
UND(400*ML,M1,2,M2,0,M3,1):: XD=XD+9 :: IF XD<>11 THEN XD=2 :: DX=11 ELSE DX=2
500 CALL COLOR(5,XD,DX):: CALL KEY(1,T,Y):: IF Y THEN 520
510 NEXT I :: NEXT II :: GOTO 460
520 CALL HCHAR(6,1,116,160):: CALL CHARSET :: FOR I=0 TO 8 :: CALL COLOR(I,16,13
):: NEXT I :: CALL COLOR(11,5,4,12,13,4):: CALL DELSPRITE(ALL)
530 DISPLAY AT(2,1):"SCORExxxxxxxxxLEVELxxxxMENxx"
540 DISPLAY AT(2,7)SIZE(7):USING "#######":PU :: DISPLAY AT(2,20)SIZE(3):R+1 ::
DISPLAY AT(2,27)SIZE(2):ME
550 FOR I=1 TO 31 STEP 2 :: CALL HCHAR(20,I,118):: CALL HCHAR(20,I+1,119):: NEXT
 I :: CALL HCHAR(21,1,117,128)
560 CALL SPRITE(#4,132,3,150,100):: CALL SPRITE(#5,128,3,150,132)
570 CALL CHAR(136,"30181830306060301818306030180C18"&RPT$("0",32))
580 CALL HCHAR(12,30,122):: CALL HCHAR(12,31,120,2):: CALL HCHAR(13,31,122):: CA
LL HCHAR(13,32,123)
590 Z=INT(RND*61)+140 :: CALL SPRITE(#2,136,13,27,Z):: CALL SPRITE(#3,136,13,59,
Z)
600 CALL SPRITE(#1,140,2,57,230)
610 SBW=(INT(RND*2)+1)*20-30 :: CALL MOTION(#2,0,SBW,#3,0,SBW):: CALL SOUND(100,
400,0,2000,1,110,1)
620 CALL POSITION(#2,SPZ,SPS):: IF (SBW*17+30)+(SPS*SBW*-.1)<=0 THEN SBW=SBW*-1
:: CALL MOTION(#2,0,SBW,#3,0,SBW)
630 CALL KEY(1,T,Y):: IF Y=0 THEN 620
640 FOR I=-11 TO 19 STEP 2 :: CALL MOTION(#1,I,-8):: IF I=-7 THEN CALL CHAR(140,
T$(2))
650 CALL POSITION(#2,SPZ,SPS):: IF (SBW*17+30)+(SPS*SBW*-.1)<=0 THEN SBW=SBW*-1
:: CALL MOTION(#2,0,SBW,#3,0,SBW)
660 CALL COINC(ALL,FG):: IF FG THEN CALL MOTION(#1,0,0,#2,0,0,#3,0,0):: GOTO 780
670 NEXT I :: CALL MOTION(#1,15,0):: FOR I=1 TO 4 :: CALL COINC(ALL,FG):: IF FG
THEN CALL MOTION(#1,0,0,#2,0,0,#3,0,0):: GOTO 780
680 NEXT I :: CALL MOTION(#2,0,0,#3,0,0)
690 CALL POSITION(#1,Z,S):: CALL SOUND(-100,300+Z*4,1):: IF Z<155 THEN 690 :: CA
```

```
LL MOTION(#1,0,0,#5,0,8,#4,0,8)
700 XD=XD+220 :: IF XD>220 THEN XD=0
710 CALL SOUND(-100,200+XD,1):: CALL COINC(#1,#5,15,FG):: IF FG THEN CALL DELSPR
ITE(#1):: CALL MOTION(#5,0,0,#4,0,0)ELSE 700
720 CALL SOUND(200,1300,2,110,1,800,3,-1,0)
730 ME=ME-1 :: DISPLAY AT(2,27)SIZE(2):ME :: IF ME=0 THEN 760 ELSE CALL DELSPRIT
E(#6,#7):: CALL CHAR(140,T$(1)):: CALL HCHAR(12,1,116,64)
740 CALL POSITION(#4,Z,S):: S=S+32 :: IF S>256 THEN S=S-256
750 CALL LOCATE(#5,Z,S):: GOTO 580
760 DISPLAY AT(3,3)SIZE(23):"PRESS BUTTON TO BEGIN !"
770 CALL KEY(1,T,Y):: IF Y=0 THEN 770 ELSE CALL DELSPRITE(ALL):: CALL HCHAR(1,1,
116,768):: GOTO 240
780 FOR I=10 TO 30 STEP 10 :: CALL SOUND(100,I*20+110,0):: NEXT I :: PU=PU+10 ::
 DISPLAY AT(2,7)SIZE(7):USING "#######":PU
790 CALL CHAR(140,T$(2)):: FOR L=1 TO 3+(R*2):: Z=INT(RND*61)+40 :: CALL SPRITE(
#6,136,13,27,Z):: CALL SPRITE(#7,136,13,59,Z)
800 H=MIN(L,4):: SBW2=(INT(RND*2)+1)*20-30 :: SBW2=SBW2+H*SGN(SBW2):: CALL MOTIO
N(#6,0,SBW2,#7,0,SBW2)
810 SBW=MIN(ABS(SBW),13)*SGN(SBW):: CALL MOTION(#1,0,SBW,#2,0,SBW,#3,0,SBW)
820 CALL POSITION(#6,SPZ,SPS):: IF (SGN(SBW2)*70+30)+(SPS*SGN(SBW2)*-1)<=0 THEN
SBW2=SBW2*-1 :: CALL MOTION(#6,0,SBW2,#7,0,SBW2)
830 CALL POSITION(#2,SPZ,SPS):: IF (SGN(SBW)*170+30)+(SPS*SGN(SBW)*-1)<=0 THEN S
BW=SBW*-1 :: CALL MOTION(#1,0,SBW,#2,0,SBW,#3,0,SBW)
840 CALL KEY(1,T,Y):: IF Y=0 THEN 820
850 CALL DELSPRITE(#3,#2):: FOR I=-13 TO 21 STEP 2 :: CALL MOTION(#1,I,(8+L)*SGN
(SBW))
860 CALL POSITION(#6,SPZ,SPS):: IF ((SBW2-L)*7+30)+(SPS*(SBW2-L)*-.1)<=0 THEN SB
W2=SBW2*-1 :: CALL MOTION(#6,0,SBW2,#7,0,SBW2)
870 CALL COINC(ALL,FG):: IF FG THEN CALL MOTION(#1,0,0,#6,0,0,#7,0,0):: GOTO 890
880 NEXT I :: CALL MOTION(#6,0,0,#7,0,0,#1,15,0):: GOTO 690
890 FOR I=10 TO 30 STEP 10 :: CALL SOUND(100,I*20+110,0):: NEXT I :: PU=PU+L*10+
(R+1)*10 :: DISPLAY AT(2,7)SIZE(7):USING "#######":PU
900 IF L=1 THEN CALL HCHAR(12,30,116,67)
910 IF L=3+(R*2)THEN 970
920 CALL MOTION(#1,0,10,#6,0,10,#7,0,10,#4,0,10,#5,0,10):: FOR I=1 TO 608 :: NEX
T I
930 CALL MOTION(#1,0,0,#6,0,0,#7,0,0,#4,0,0,#5,0,0)
940 CALL POSITION(#6,Z,S):: CALL SPRITE(#2,136,13,Z,S):: CALL SPRITE(#3,136,13,Z
+32,S):: SBW=SBW2
950 CALL POSITION(#4,Z,S):: S=S+32 :: IF S>256 THEN S=S-256
960 CALL LOCATE(#5,Z,S):: NEXT L
970 CALL HCHAR(12,1,120,2):: CALL HCHAR(12,3,123):: CALL HCHAR(13,2,121):: CALL
HCHAR(13,1,122)
980 CALL MOTION(#1,0,SBW2,#6,0,SBW2,#7,0,SBW2)
990 CALL POSITION(#6,SPZ,SPS):: IF (SGN(SBW2)*70+30)+(SPS*SGN(SBW2)*-1)<=0 THEN
SBW2=SBW2*-1 :: CALL MOTION(#1,0,SBW2,#6,0,SBW2,#7,0,SBW2)
1000 CALL KEY(1,T,Y):: IF Y=0 THEN 990
1010 CALL MOTION(#6,0,0,#7,0,0):: FOR I=-11 TO 19 STEP 2 :: CALL MOTION(#1,I,(8+
L)*SGN(SBW2))
1020 CALL POSITION(#1,Z,S):: IF S>200 OR(Z>76 AND Z<200)THEN 1050 ELSE IF Z>60 A
ND S<17 THEN 1070
1030 NEXT I :: CALL MOTION(#1,15,0):: FOR I=1 TO 10 :: CALL POSITION(#1,Z,S):: I
F S>16 THEN 1050 ELSE IF Z>60 AND Z<80 THEN 1070
1040 NEXT I
1050 IF S>200 THEN CALL MOTION(#1,0,0):: CALL LOCATE(#1,Z,1)
1060 CALL MOTION(#1,15,0):: GOTO 690
1070 PU=PU+100*(R+1):: DISPLAY AT(2,7)SIZE(7):USING "#######":PU :: CALL MOTION(
#1,0,0):: CALL CHAR(140,T$(1)):: CALL LOCATE(#1,60,5)
1080 A=-3 :: FOR I=40 TO 10 STEP -2 :: FOR II=-10*I TO 10*I STEP 50 :: CALL SOUN
D(-100,II+610,2):: A=A+8 :: IF A>13 THEN A=5
1090 CALL COLOR(11,A,4,12,A,4):: NEXT II :: NEXT I :: CALL COLOR(11,5,4,12,13,4)
1100 CALL DELSPRITE(#1,#6,#7):: CALL HCHAR(12,1,116,35)
1110 CALL CHAR(140,"060F130909060D15157D110A040A091B"&RPT$("0",32))
1120 CALL CHAR(136,"000000000000000000010708110609130000000000000000002040F088C4B0
C8F4")
1130 CALL CHAR(112,"000000000000000000003070E1F1B0D0F070000000000000000000C0E0B070F0
E")
1140 CALL CHAR(108,"000000000000000000014B370505090100000000000000000018FCF8E02010
101")
1150 FOR I=1 TO 28 STEP 3 :: CALL HCHAR(14,I,123):: CALL HCHAR(14,I+1,122):: CAL
L HCHAR(14,I+2,121):: NEXT I
1160 CALL HCHAR(14,31,123):: CALL HCHAR(14,32,122)
1170 KM=0 :: GS=MIN(-1,R-6):: CALL SPRITE(#1,140,2,73,230,0,GS)
1180 RANDOMIZE :: G=R*2+15 :: IF G>30 THEN G=30
1190 Z=INT(RND*3)
```

```
1200 IF Z=0 THEN CALL SPRITE(#2,136,13,73,1,0,G)ELSE IF Z=1 THEN CALL SPRITE(#2,
112,15,73,1,0,G)ELSE CALL SPRITE(#2,108,9,73,1,0,G)
1210 KM=KM+1 :: CALL COINC(ALL,FG):: IF FG THEN 1250 ELSE CALL POSITION(#2,Z,S):
: IF S>220 THEN 1190
1220 CALL POSITION(#1,Z,S):: IF S<10 THEN 1370
1230 CALL KEY(1,T,Y):: IF Y<>0 THEN 1330
1240 CALL COINC(ALL,FG):: IF FG=0 THEN 1210
1250 CALL MOTION(#1,15,0,#2,0,0)
1260 CALL POSITION(#1,Z,S):: CALL SOUND(-100,300+Z*4,1):: IF Z<155 THEN 1260 ::
CALL MOTION(#1,0,0,#5,0,8,#4,0,8)
1270 XD=XD+220 :: IF XD>220 THEN XD=0
1280 CALL SOUND(-100,200+XD,1):: CALL COINC(#1,#5,15,FG):: IF FG THEN CALL DELSP
RITE(#1):: CALL MOTION(#5,0,0,#4,0,0)ELSE 1270
1290 CALL SOUND(200,1300,2,110,1,800,3,-1,0)
1300 PU=PU+KM :: DISPLAY AT(2,7)SIZE(7):USING "#######":PU
1310 ME=ME-1 :: DISPLAY AT(2,27)SIZE(2):ME :: IF ME=0 THEN 760 ELSE CALL POSITIO
N(#4,Z,S):: S=S+32 :: IF S>256 THEN S=S-256
1320 CALL LOCATE(#5,Z,S):: GOTO 1170
1330 FOR I=-17 TO 17 STEP 2 :: CALL MOTION(#1,I,GS):: KM=KM+1
1340 CALL POSITION(#1,Z,S):: IF S<10 THEN 1370
1350 CALL COINC(ALL,FG):: IF FG THEN 1250
1360 NEXT I :: CALL MOTION(#1,0,GS):: GOTO 1210
1370 CALL MOTION(#1,0,0,#2,0,0):: PU=PU+KM+(R+1)*20 :: DISPLAY AT(2,7)SIZE(7):US
ING "#######":PU
1380 FOR II=1 TO 2 :: FOR I=1 TO 3 :: CALL SOUND(100,I*200+110,0):: NEXT I :: FO
R I=1 TO 30 :: NEXT I :: NEXT II
1390 FOR I=1 TO 30 :: NEXT I :: FOR I=1 TO 6 :: CALL SOUND(100+I*10,I*200+110,0)
:: NEXT I
1400 CALL DELSPRITE(#1,#2):: CALL CHAR(136,"0000000000000000000000030F1F2444480000
0000000000000000080E0F0484424")
1410 CALL CHAR(112,"60F0C8D0D060B8ACA8A8887020202038"&RPT$("0",32))
1420 CALL HCHAR(11,5,122):: CALL VCHAR(11,6,120,3):: CALL HCHAR(11,7,121):: CALL
 HCHAR(14,1,116,5)
1430 CALL HCHAR(9,1,120,2):: CALL HCHAR(9,3,123):: CALL HCHAR(10,2,121):: CALL H
CHAR(10,1,122)
1440 CALL SPRITE(#6,112,12,33,6):: RANDOMIZE :: CALL SPRITE(#2,136,14,73,INT(RND
*80)+26):: CALL SPRITE(#3,136,14,73,INT(RND*80)+126)
1450 CALL SPRITE(#1,140,2,73,235,0,0)
1460 SBW=(INT(RND*2)+1)*20-30 :: SBW=SBW+R*2*SGN(SBW):: SBW2=(INT(RND*2)+1)*20-3
0 :: SBW2=SBW2+R*2*SGN(SBW2):: CALL MOTION(#2,0,SBW,#3,0,SBW2)
1470 CALL POSITION(#2,Z,S):: IF 10*SGN(SBW)*6.8+32+S*SGN(SBW)*-1<=0 THEN SBW=SBW
*-1 :: CALL MOTION(#2,0,SBW)
1480 CALL POSITION(#3,Z,S):: IF 10*SGN(SBW2)*17.25+22.5+S*SGN(SBW2)*-1<=0 THEN S
BW2=SBW2*-1 :: CALL MOTION(#3,0,SBW2)
1490 CALL COINC(ALL,FG):: IF FG THEN 1580
1500 CALL JOYST(1,T,Y):: CALL MOTION(#1,0,T*2-(R*T/2)):: CALL KEY(1,T,Y):: IF Y<
>0 THEN 1520
1510 CALL POSITION(#1,Z,S):: IF S<36 THEN 1580 ELSE CALL COINC(ALL,FG):: IF FG T
HEN 1580 ELSE 1470
1520 FOR I=-3 TO 3 :: CALL MOTION(#1,I^3,-10-R*2)
1530 CALL POSITION(#2,Z,S):: IF 10*SGN(SBW)*6.8+32+S*SGN(SBW)*-1<=0 THEN SBW=SBW
*-1 :: CALL MOTION(#2,0,SBW)
1540 CALL POSITION(#3,Z,S):: IF 10*SGN(SBW2)*17.25+22.5+S*SGN(SBW2)*-1<=0 THEN S
BW2=SBW2*-1 :: CALL MOTION(#3,0,SBW2)
1550 CALL COINC(ALL,FG):: IF FG THEN 1580
1560 IF I=0 THEN CALL POSITION(#1,Z,S):: IF S<26 OR S>50 THEN 1570 ELSE 1640
1570 NEXT I :: CALL MOTION(#1,0,0):: CALL COINC(ALL,FG):: IF FG=0 THEN CALL POSI
TION(#1,Z,S):: CALL LOCATE(#1,73,S):: GOTO 1470
1580 CALL MOTION(#1,15,0,#2,0,0,#3,0,0)
1590 CALL POSITION(#1,Z,S):: CALL SOUND(-100,300+Z*4,1):: IF Z<155 THEN 1590 ::
CALL MOTION(#1,0,0,#5,0,8,#4,0,8)
1600 XD=XD+220 :: IF XD>220 THEN XD=0
1610 CALL SOUND(-100,200+XD,1):: CALL COINC(#1,#5,15,FG):: IF FG THEN CALL DELSP
RITE(#1):: CALL MOTION(#5,0,0,#4,0,0)ELSE 1600
1620 CALL SOUND(200,1300,2,110,1,800,3,-1,0):: PU=PU+30*(R+1):: DISPLAY AT(2,7)S
IZE(7):USING "#######":PU
1630 ME=ME-1 :: DISPLAY AT(2,27)SIZE(2):ME :: IF ME=0 THEN 760 ELSE 1440
1640 CALL MOTION(#1,0,0,#2,0,0,#3,0,0):: CALL COLOR(#2,7,#3,7):: XD=7
1650 CALL CHAR(136,"00001C3E7F7F7F7F3F1F0F0703010000000387CFEFEFEFEFCF8F0E0C080
0000")
1660 PU=PU+330*(R+1):: DISPLAY AT(2,7)SIZE(7):USING "#######":PU :: ME=ME+1 :: D
ISPLAY AT(2,27)SIZE(2):ME
1670 DATA 1,523,440,349,1,523,392,349,1,523,392,330,1,440,440,349,1,523,440,349,
1,523,392,349,1,523,392,330
1680 DATA 1,440,440,349,1,587,349,247,1,523,320,262,1,440,349,294,1,392,392,330,
```

```
1,440,330,262,2,349,349,349
1690 RESTORE 1670 :: FOR I=1 TO 14 :: READ ML,M1,M2,M3 :: CALL SOUND(400*ML,M1,2
,M2,0,M3,1):: XD=XD+3 :: IF XD>10 THEN XD=7
1700 CALL COLOR(#2,XD,#3,XD):: NEXT I
1710 DATA 1,440,440,349,1,523,392,330,1,587,349,294,1,523,392,330,1,523,392,330,
1,698,440,349,1,698,494,392
1720 DATA 2,523,440,349
1730 FOR II=1 TO 2 :: RESTORE 1710 :: FOR I=1 TO 8 :: READ ML,M1,M2,M3 :: CALL S
OUND(400*ML,M1,2,M2,0,M3,1):: XD=XD+3 :: IF XD>10 THEN XD=7
1740 CALL COLOR(#2,XD,#3,XD):: NEXT I :: NEXT II
1750 R=R+1 :: DISPLAY AT(2,20)SIZE(3):R+1 :: CALL DELSPRITE(#6,#1,#2,#3):: CALL
HCHAR(9,1,116,224):: GOTO 570
```

TIPS + TRICKS

Schneider CPC-464

Interne Abspeicherung von BASIC-Zeilen

Das nachfolgende Programm behandelt Tips und Tricks für den Schneider CPC-464 und besteht aus zwei Teilen. Der Teil 1 von Zeile 20 bis Zeile 60 zeigt den Speicherinhalt Ihres Computers in folgenden Formen: Hex-Adresse, Adresseninhalt hex, und wenn der Speicherzelleninhalt größer als 32 (dez) ist, auch den Zeichensatz des CPC-464.

Die Werte, die kleiner als 32 sind, dürfen nicht mit PRINT CHR$ (...) ausgegeben werden, da diese sonst als Steuerzeichen aufgefaßt werden (siehe Handbuch, Kapitel 9, Seite 2–4).
Zeile 10 wird für Programmteil 2 benötigt und sollte genauso eingegeben werden, wie es beschrieben wurde. Nach dem Befehl REM zwei (!) Leerräume und dann ein X.
In Zeile 20 können die Werte für die Schleife auch anders gewählt werden (max. von 0 bis 2↑16).
Nach RUN sieht man, daß das Basic-Programm bei Adresse 367 beginnt.
Von Adresse 368 bis Adresse 375 ist die Zeile 10 des Programms folgendermaßen abgelegt:

Adresse	Speicherinhalt (jew. dezimal)
367 0	Kennzeichnung Basic-Anfang
368 9	(Zeilenlänge LOW-Byte)
369 0	(Zeilenlänge HIGH-Byte)
370 10	(Zeilennummer LOW-Byte)
371 0	(Zeilennummer HIGH-Byte)
372 197	REM (siehe weiter unten)
373 32	Leerraum
374 32	Leerraum
375 120	x

Wie aus dem Beispiel zu erkennen ist, benötigt der CPC zwei Bytes (368 LOW/369 HIGH), um die Zeilenlänge

Basic-Befehle mit Dezimal-Code (ToKens)

Dezimal-Code von 0 bis 127 siehe Handbuch CPC 464 Anhang 3/1

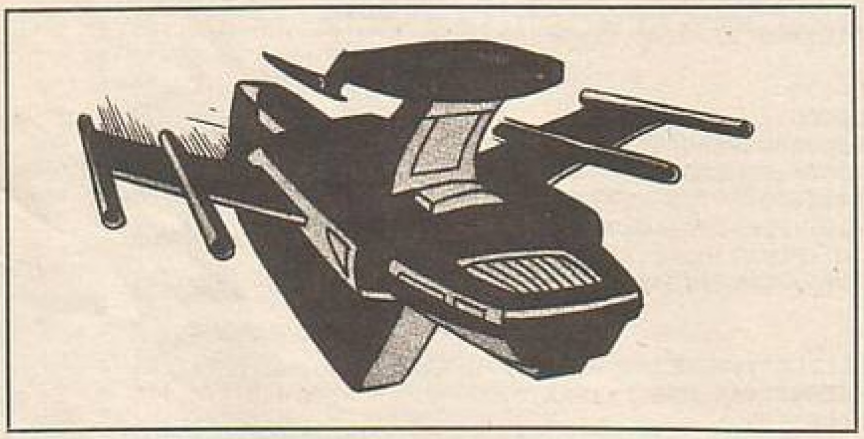
Basic-Befehle mit 1-Byte-Dezimal-Code

128 AFTER	162 INK	196 RELEASE
129 AUTO	163 INPUT	197 REM
130 BORDER	164 KEY	198 RENUM
131 CALL	165 LET	199 RESTORE
132 CAT	166 LINE	200 RESUME
133 CHAIN	167 LIST	201 RETURN
134 CLEAR	168 LOAD	202 RUN
135 CLG	169 LOCATE	203 SAVE
136 CLOSEIN	170 MEMORY	204 SOUND
137 CLOSEOUT	171 MERGE	205 SPEED
138 CLS	172 MID$	206 STOP
139 CONT	173 MODE	207 SYMBOL
140 DATA	174 MOVE	208 TAG
141 DEF	175 MOVER	209 TAGOFF
142 DEFINT	176 NEXT	210 TROFF
143 DEFREAL	177 NEW	211 TRON
144 DEFSTR	178 ON	212 WAIT
145 DEG	179 ON BREAK	213 WEND
146 DELETE	180 ON BREAK GOTO	214 WHILE
147 DIM	181 ON SQ	215 WIDTH
148 DRAW	182 OPENIN	216 WINDOW
149 DRAWR	183 OPENOUT	217 WRITE
150 EDIT	184 ORIGIN	218 ZONE
151 ELSE	185 OUT	219 DI
152 END	186 PAPER	220 EI
153 ENT	187 PEN	221 syntax error
154 ENV	188 PLOT	... ...
155 ERASE	189 PLOTR	226 syntax error
156 ERROR	190 POKE	227 ERL
157 EVERY	191 PRINT	228 FN
158 FOR	192 ,	229 SPC
159 GOSUB	193 RAD	230 STEP
160 GOTO	194 RANDOMIZE	231 SWAP
161 IF	195 READ	232 syntax error

```
2100 DATA255,129,255,129,129,255,129,255:REM            FELS
2110 DATA60,126,126,60,24,24,24,24:REM            BAUM
2120 DATA56,16,56,56,16,124,170,170:REM            MONSTER
2130 DATA52,126,255,117,238,255,122,60:REM            FELSEN
2140 DATA17,4,0,9,0,186,255,59:REM            FELSEN
2990 FORI=0TO111:READV:POKE7168+0*8+I,V:NEXT
3000 POKE37154,255:POKE36869,240:POKE36879,14:PRINT"[illegible]            COLECTOR
3005 PRINT"[illegible] LEVEL 1 = MONSER ERST[illegible]            AB RAUM 5"
3010 PRINT"[illegible] LEVEL 2 = MONSTER"
3020 PRINT"[illegible] LEVEL 3 = SCHRITTE"
3030 PRINT"[illegible] LEVEL 4 = MONSTER UND            SCHRITTE"
3100 PRINT"[illegible]WELCHER LEVEL"
3110 YR=1:INPUT"[illegible]";YR
3130 IFYR=1THEN10
3140 IFYR=2THENRT=1:GOTO10
3150 IFYR=3THENJK=1:GOTO10
3160 IFYR=4THENJK=1:RT=1:GOTO10
3170 IFYR>4THEN3000
3900 REM"[illegible]
4000 IF7822+XX<7680THENXX=XX+YY
4005 IF7822+XX>8185THENXX=XX+YY
4010 RETURN
```

MOON-LANDER

TI-99/4A



Auf den Spuren von „APOLLO“:

Eine weiche Landung ist sehr wichtig!

Moon-Lander ist für den TI-99/4A mit Extended Basic und Joysticks geschrieben. Aber es ist auch noch genügend Speicherplatz zur Verfügung, so daß das Spiel auch auf eine Tastatur-Version umgeändert werden kann.

Nach dem Eintippen und nach Eingabe von „RUN“ wird man nach der Spielstärke gefragt (leicht, mittel, schwer). Dann wird der Bildschirm zunächst dunkel. Nach einiger Zeit erscheint der Bildschirmaufbau des Spieles mit der Frage nach der Spielanleitung, die mit einer 3stimmigen Melodie untermalt ist. Je nach Antwort erscheint entweder eine kurze Spielanleitung oder das Spiel beginnt sofort.

Zum Spiel:

Oben links in der Ecke erscheint Ihr Lander. Ihre Aufgabe ist es nun, diesen Lander sicher auf einen der beiden Plattformen zu landen. Doch das ist gar nicht so einfach. Infolge der Anziehungskraft sinkt Ihr Lander unaufhörlich. Sie sind somit gezwungen, gegenzusteuern. Dies geschieht folgendermaßen:

Drücken Sie den Joystickhebel nach unten, so benutzen Sie die „Bremsdüsen“, und Ihr Lander wird abgebremst. Diese Bremsdüsen kann man aber nicht sehen, sondern nur hören. Je länger Sie bremsen, desto kleiner wird die Geschwindigkeit Ihres Landers. Das gleiche gilt für die anderen Richtungen:

Hebel nach rechts: Lander fliegt nach links (die Düsen, die rechts am Lander sind, lassen den Lander nach links fliegen).

Hebel nach links: Lander fliegt nach rechts.

Hebel nach oben: Lander fliegt nach unten (er wird beschleunigt).

Zu einer erfolgreichen Landung gehört aber noch mehr. Am linken Bildschirmrand sehen Sie einen hellblauen senkrechten Streifen, in dessem Zentrum befindet sich ein grüner Balken. Dies ist die Anzeige Ihrer Geschwindigkeit. Der schwarze waagerechte Strich zeigt Ihre momentane Geschwindigkeit an. Je höher dieser Strich ist, desto kleiner ist Ihre Geschwindigkeit. Um eine erfolgreiche Landung zu erreichen, dürfen Sie die Felsen nicht berühren (sonst explodieren Sie), zusätzlich muß eine bestimmte Geschwindigkeit eingehalten werden. Der schwarze Strich muß sich im grünen Bereich oder darüber befinden. Je höher dieser Strich ist, desto mehr Punkte bekommen Sie bei einer geglückten Landung. Ist der Strich unter dem grünen Bereich, so sind Sie zu schnell gelandet und explodieren ebenfalls. Auch wenn Sie den Bildschirm nach rechts, links oder oben verlassen, explodieren Sie.

Zur Punktverteilung:

Die Anzahl der zu vergebenden Punkte ist abhängig von der Stellung des schwarzen Striches. Diese Punkte werden dann noch mit dem Multiplikator 2 oder 3 multipliziert (mit 2, wenn Sie auf der linken Plattform landen, mit 3, wenn Sie auf der rechten Plattform lan-

den, weil dort eine gute Landung schwieriger zu realisieren ist). Ganz unten am Rand sehen Sie Ihren Fuel-Vorrat. Ist dieser aufgebraucht, so ist das Spiel vorbei. Wenn Sie eine gute Punktzahl erzielt haben, dann können Sie Ihren Namen in eine Tabelle eintragen. Das geschieht folgendermaßen:
Mit dem Joystick bewegen Sie einen schwarzen „Rahmen". Wenn Sie nun auf den Feuerknopf drücken, dann haben Sie denjenigen Buchstaben eingegeben, der im schwarzen Rahmen ist. Falsch eingegebene Buchstaben können mit „R." für rub (am Ende der unteren Buchstabenreihe) gelöscht werden. Das Eintragen ist beendet, wenn so viele Buchstaben eingegeben wurden, wie möglich sind, oder wenn Sie das Zeichen „E." für End eingegeben haben (auch am Ende der unteren Buchstabenreihe). Die jeweils 5 besten Ergebnisse sind in dieser Tabelle gespeichert.

Fuelvorrat:

Sie verlieren Fuel, wenn Sie die Bremsdüsen betätigen oder wenn Sie explodieren. Bei einer Explosion ist der Fuel-Verlust von der jeweiligen Spielstärke abhängig. Bei einer schwierigeren Spielstärke verlieren Sie dann mehr Fuel als bei einer leichteren. Sie bekommen Fuel dazu, wenn Sie glücklich gelandet sind. Je mehr Punkte Sie für eine Landung bekommen, desto mehr Fuel erhalten Sie auch dafür. Je mehr Punkte Sie nun bekommen (je länger Sie also spielen), desto schwieriger wird das Spiel, denn mit steigender Punktzahl verschlechtert sich die Bremskraft der Düsen und man muß häufiger und länger bremsen.
Nun viel Spaß und Erfolg bei Moon-Lander.

Auf geht's: Ohne Fleiß, kein „Preis"!

```
10 !************************
20 !*      MOON-LANDER      *
30 !*        (C)1984        *
40 !*  BY HEYO SPEKKER      *
50 !*  BUCHENSTR.16         *
60 !*  2913 AUGUSTFEHN      *
70 !************************
80 DATA 587,349,117,587,349,117,587,349,117,554,330,117,587,370,147,587,370,147,
587,370,147,523,370,147
90 DATA 466,392,156,466,392,156,466,392,156,392,311,156,349,294,117,349,294,117,
349,294,117,440,349,117
100 DATA 466,392,156,466,392,156,466,392,156,392,311,156,349,294,117,349,294,117
,349,294,117,587,349,117
110 DATA 523,330,131,392,330,131,440,262,131,466,294,131,523,311,175,523,311,175
,698,440,110,622,392,110
120 DATA 587,349,117,587,349,117,587,349,117,554,330,117,587,370,147,587,370,147
,587,370,147,523,370,147
140 DATA 466,392,156,466,392,156,466,392,156,523,370,156,587,370,147,587,370,147
,587,370,147,523,370,147
150 DATA 466,392,156,466,392,156,466,392,156,392,311,156,349,294,233,370,262,220
,392,247,196,392,349,196
160 DATA 587,330,131,587,330,131,523,311,175,523,311,175,466,294,117,466,15000,1
5000,523,15000,15000,554,330,15000
170 DATA 587,370,15000,15000,15000,147,587,370,147,587,370,147,466,294,196,440,2
62,196,392,233,196,392,233,196
180 DATA 587,370,15000,587,370,15000,15000,15000,147,523,15000,15000,466,294,196
,440,262,196,392,233,196,392,233,196
190 DATA 523,330,15000,15000,15000,131,523,330,131,523,330,131,440,262,175,440,2
62,175,349,220,175,349,220,175
200 DATA 523,330,131,523,15000,15000,587,15000,15000,523,330,131,698,440,175,150
00,15000,15000,698,440,175,622,392,175
310 AA=0 :: CALL BILD(PP)
320 SP=32 :: CALL SCREEN(2):: CALL CLEAR :: CALL MAGNIFY(4)
330 DEF F=(D(2)-D(1))*4
340 CALL CHAR(81,"3C4299A1A199423C")
350 CALL CHAR(36,"0007010F1F3F797F7F7F7F7F3F08102000E080F0F8FC9EFEFEFEFEFEFEFC1008
04")
360 CALL CHAR(40,"FF000000000000000000000000000000FF000000000000000000")
370 CALL CHAR(32,"0000000000000000FFFFFFFFFFFFFFFF0000000000000000FFFFFFFFFFFFFF
FF")
380 CALL CHAR(92,"80603C1F0F1F7C3B0B0D0E1F1930200040E0E0F8FF7EB8D8BCDE59B8FCFC8E
06")
390 CALL CHAR(64,"0024187E18240000")
400 DATA 0000003CFFFFFFFFFFFFFFFFFFFFFFFF00000000080C0C0FCFEFEFFFFFFFFFF
410 DATA FFFFFFFFFFFFFFFFFFFFFFFEFEFCFC8080C0C0C0C0C0808080808080800000000
420 DATA FCF8F0E0C0C0C0C0C0C0E0E0F0F8FCFF000000000000000000000000000000FF
430 DATA 000000000000000000000000000000FF3F3F7F7F3F3F3F3F3F7F7F3F3F3F7FFF
440 DATA 1F1F0F0F1F1F0F0F0F0F0F1F1F1F3F3FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
450 DATA 0101030303070F0F0F0707070F0F1FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
460 DATA 000000000000000000000000000000010103030707 0F1F1F1F3F3F7F7F7FFF
470 DATA 0000000000000000103070F1F3F7FFF0000071F1F3F7FFFFFFFFFFFFFFFFFFFF
480 DATA 0000FFFFFFFFFFFFFFFFFFFFFFFFFFFF00C0E0F0F0F0F0F8F8F8F8FCFCFCFCFC
490 DATA FCFCF8F8F0F0E0C0C0808080808080FF000000000000000000000000000000FF
500 DATA 0000000101030303030303070F0FFF3F7FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
510 DATA 0103070F0F0F1F3F3F3F3F3F3F3F3F3FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

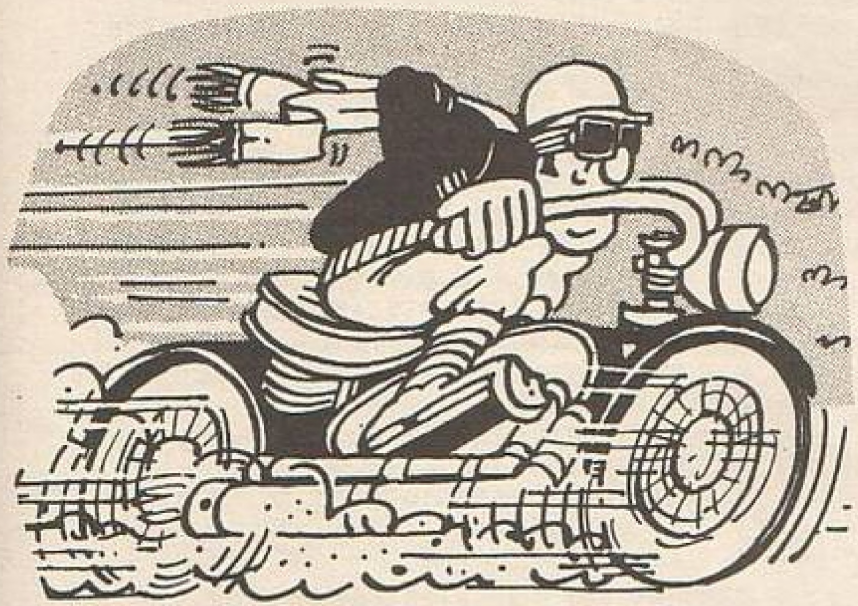
```
520 DATA 0000000000000000103070F1F3F7FFF0303070F1F3F7FFFFFFFFFFFFFFFFFFFF
530 DATA 13,2,17,4,21,4,21,8,17,10,13,10,9,10,5,14,5,18,9,20,9,24,5,26,1,28
540 DATA 17,2,21,12,9,14,9,16,9,18,13,20,13,22,13,24,13,26,9,28,5,30
550 CALL CHAR(88,""):: CALL COLOR(8,2,2):: FOR I=3 TO 7 :: CALL COLOR(I,16,2):: 
NEXT I
560 RESTORE 400 :: FOR I=96 TO 140 STEP 4 :: READ A$ :: CALL CHAR(I,A$):: NEXT I
 :: READ A$ :: CALL CHAR(44,A$)
570 A=96 :: FOR I=1 TO 13 :: READ B,C :: CALL SPRITE(#I,A,2,B*8-7,C*8-7):: A=A+4
 :: IF A=144 THEN A=44
580 NEXT I
590 FOR I=1 TO 11 :: READ B,C :: CALL VCHAR(B,C,88,25-B):: CALL VCHAR(B,C+1,88,2
5-B):: NEXT I :: CALL VCHAR(1,32,88,24)
600 FOR I=1 TO 13 :: CALL COLOR(#I,7):: NEXT I :: CALL COLOR(8,7,7)
610 CALL COLOR(9,8,8,11,3,3,12,11,11)
620 CALL VCHAR(11,32,97,12):: CALL VCHAR(13,32,113,2)
630 DISPLAY AT(24,10):"FUEL" :: CALL HCHAR(24,16,127,17):: DISPLAY AT(22,14)SIZE
(9):"TOP";TOP :: DISPLAY AT(20,14)SIZE(9):"PUNKTE";PT
640 CALL SPRITE(#16,32,7,161,233)
650 IF AA=1 THEN 690
660 DISPLAY AT(1,3):"MOON LANDER" :: DISPLAY AT(3,3):"©1984 H SPEKKER" :: DISPLA
Y AT(15,12)SIZE(18):"SPIELREGELN?(J/N)"
670 GOTO 1300
680 CALL HCHAR(1,3,32,20):: CALL HCHAR(3,3,32,20):: CALL HCHAR(15,12,88,20):: FO
R I=1 TO 500 :: NEXT I
690 D(1),E(1),E(2),AA=0 :: CALL SPRITE(#14,36,5,1,9,#15,40,2,99,249)
700 CALL MOTION(#16,0,0):: CALL JOYST(1,X,Y):: IF X=0 AND Y=0 THEN 710 ELSE 720
710 CALL SOUND(-2000,-7,9):: D(2)=D(1)+0.5 :: GOTO 740
720 CALL SOUND(-2000,-6,9):: D(2)=D(1)+Y/INT(PT/500+8):: E(2)=E(1)-X/INT(PT/500+
8):: CALL MOTION(#16,0,-1):: ZA=ZA+1 :: IF ZA/4=INT(ZA/4)THEN 725 ELSE 730
725 CALL HCHAR(24,SP,88):: SP=SP-1
730 CALL POSITION(#16,B,C):: IF C<=121 OR SP<=16 THEN 1330
740 CALL MOTION(#14,D(2),E(2),#15,F,0):: CALL COINC(ALL,C):: IF C THEN CALL CRAS
H(SP,PP):: GOTO 690
750 D(1)=D(2):: E(1)=E(2):: CALL POSITION(#15,G,H):: IF G<72 OR G>176 THEN CALL
MOTION(#15,0,0)
760 CALL POSITION(#14,Q,W):: IF W>254 THEN CALL CRASH(SP,PP):: GOTO 690 ELSE IF
Q>154 AND Q<159 AND W<=65 AND W>=33 THEN ZU=2 :: GOTO 800
770 CALL COINC(ALL,C):: IF C THEN CALL CRASH(SP,PP):: GOTO 690 ELSE IF Q>=59 AND
 Q<=73 AND W<179 AND W>154 THEN ZU=3 :: GOTO 800
780 IF Q>193 AND Q<230 THEN GOSUB 1100 :: GOTO 690
790 CALL COINC(ALL,C):: IF C THEN CALL CRASH(SP,PP):: GOTO 690 ELSE 700
800 CALL MOTION(#14,0,0,#15,0,0,#16,0,0):: CALL POSITION(#15,G,H):: IF G<=112 TH
EN CALL MELODY1(ZU,SP,PT)ELSE CALL MELODY2 :: CALL CRASH(SP,PP)
810 GOTO 690
820 Z$="" :: CALL DELSPRITE(ALL):: CALL CLEAR :: CALL SCREEN(11)
830 CALL CHARSET :: CALL CHAR(91,"0070507050484A"):: CALL CHAR(92,"0078407840407
A")
840 FOR I=2 TO 8 :: CALL COLOR(I,2,11):: NEXT I :: CALL CHAR(93,"FF818181818181F
F")
850 IF PT>P(5)THEN IF PT>P(4)THEN 860 ELSE P(5)=PT :: B1=5 B$(5)="" ELSE 1090
860 IF PT>P(3)THEN IF PT>P(2)THEN 870 ELSE 880 ELSE 890
870 IF PT>P(1)THEN 900 ELSE 910
880 FOR I=5 TO 4 STEP -1 :: P(I)=P(I-1):: B$(I)=B$(I-1):: NEXT I :: P(3)=PT :: B
$(3)="" :: B1=3 :: GOTO 920
890 P(5)=P(4):: B$(5)=B$(4):: P(4)=PT :: B$(4)="" :: B1=4 :: GOTO 920
900 FOR I=5 TO 2 STEP -1 :: P(I)=P(I-1):: B$(I)=B$(I-1):: NEXT I :: P(1)=PT :: B
$(1)="" :: B1=1 :: GOTO 920
910 FOR I=5 TO 3 STEP -1 :: P(I)=P(I-1):: B$(I)=B$(I-1):: NEXT I :: P(2)=PT :: B
$(2)="" :: B1=2
920 DISPLAY AT(1,9):"NAME REGISTRATION"
930 DISPLAY AT(5,6):"A B C D E F G H I J" :: DISPLAY AT(7,6):"K L M N O P Q R S
T" :: DISPLAY AT(9,6):"U V W X Y Z .   [ \"
940 CALL MAGNIFY(2):: CALL SPRITE(#1,93,2,28,52)
950 X=15 :: FOR I=1 TO 5 :: DISPLAY AT(X,6):I :: DISPLAY AT(X,8):P(I):: DISPLAY
AT(X,15):B$(I):: X=X+2 :: NEXT I
960 DISPLAY AT(11,6):"YOUR NAME:" :: CALL HCHAR(11,18,95,12)
970 X1=28 :: Y1=52 :: A1=18
980 CALL JOYST(1,X,Y):: IF X=0 AND Y=0 THEN 1030
990 IF X=-4 THEN Y1=Y1-16 ELSE IF X=4 THEN Y1=Y1+16
1000 IF Y1=36 THEN Y1=196 :: X1=X1-16 :: IF X1=12 THEN X1=60 :: GOTO 1020
1010 IF Y1=212 THEN Y1=52 :: X1=X1+16 :: IF X1=76 THEN X1=28
1020 CALL LOCATE(#1,X1,Y1)
1030 CALL KEY(1,T,S):: IF S=0 THEN 980
1040 CALL GCHAR((X1+12)/8,(Y1+12)/8,Z1):: IF Z1=92 THEN 1080
1050 IF Z1=91 THEN Z1=95 :: A1=A1-1 :: IF A1=17 THEN A1=18 :: GOTO 1030
```

(weiter S. 42)

Fortsetzung von Seite 40

```
1060 CALL HCHAR(11,A1,Z1):: IF Z1<>95 THEN Z$=Z$&CHR$(Z1):: A1=A1+1 :: IF A1=30
THEN 1080 ELSE 980 ELSE 1070
1070 Z$=SEG$(Z$,1,LEN(Z$)-1):: GOTO 980
1080 TOP=P(1):: B$(B1)=Z$ :: DISPLAY AT(13+B1*2,15)BEEP:Z$ :: FOR I=1 TO 1000 ::
 NEXT I
1090 A$="" :: PT,ZA=0 :: CALL DELSPRITE(ALL):: CALL CLEAR :: RETURN
1100 CALL CRASH(SP,PP):: DISPLAY AT(1,3)BEEP:"OUT OF ATMOSPHERE!" :: FOR I=1 TO
500 :: NEXT I :: CALL HCHAR(1,3,32,22):: RETURN
1110 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL SCREEN(11):: CALL CHARSET :: FOR I
=2 TO 8 :: CALL COLOR(I,2,11):: NEXT I
1120 PRINT "        SPIELANLEITUNG                  ---------------",,
1130 PRINT "SINN DES SPIELES IST ES,DEN",,:: PRINT "LANDER AUF DEN PLATTFORMEN",
,:: PRINT "DER KRATER ZU LANDEN.DABEI",,
1140 PRINT "MUSS EINE BESTIMMTE GESCHWIN",,:: PRINT "DIGKEIT EINGEHALTEN WERDEN.
",,:: PRINT "(DER SCHWARZE GESCHWINDIG-",,
1150 PRINT "KEITSANZEIGER MUSS SICH IM",,:: PRINT "GRUENEN BEREICH ODER DARUE-",
,
1160 PRINT "BER BEFINDEN.)JE GERINGER",,:: PRINT "DIE GESCHWINDIGKEIT,",,,,:: DI
SPLAY AT(24,10):"WEITER MIT ENTER >>"
1170 CALL KEY(3,K,S):: IF K<>13 THEN 1170
1180 CALL CLEAR :: PRINT "DESTO GROESSER DIE PUNKTZAHL",,:: PRINT "UND DESTO MEH
R FUEL BEKOMMT",,
1190 PRINT "MAN DAZU.BEI EINER NICHT GE-",,:: PRINT "GLUECKTEN LANDUNG ODER BEI"
,,
1200 PRINT "BERUEHRUNG DER FELSEN ER-",,:: PRINT "FOLGT EIN CRASH,UND MAN BE-",,
:: PRINT "KOMMT FUEL ABGEZOGEN.",,,
1210 PRINT "STEUERUNG:" :: PRINT "---------",,:: PRINT "JOYST.(L)=LANDER NACH RE
CHTS",,
1220 PRINT "JOYST.(R)=LANDER NACH LINKS",,,,,,,,:: DISPLAY AT(24,26):">>"
1230 CALL KEY(3,K,S):: IF K<>13 THEN 1230
1240 CALL CLEAR :: PRINT "JOYST(U)=LANDER NACH OBEN",,:: PRINT "JOYST(O)=LANDER
NACH UNTEN",,
1250 PRINT "DAS SPIEL IST ZU ENDE,WENN",,:: PRINT "DER FUEL AUFGEBRAUCHT IST.",,
1260 PRINT "GUTE ERGEBNISSE KOENNEN AM",,:: PRINT "SCHLUSS IN EINER TABELLE EIN"
,,:: PRINT "GETRAGEN WERDEN.DIES ER-",,
1270 PRINT "FOLGT DURCH BEWEGEN DES JOY-",,:: PRINT "STICKS UND DURCH DRUECKEN",
,:: PRINT "DES FEUERKNOPFES.VIEL SPASS!": : : .
1280 DISPLAY AT(24,10):"START MIT ENTER >>"
1290 CALL KEY(3,K,S):: IF K<>13 THEN 1290 ELSE AA=1 :: K=0 :: GOTO 320
1300 RESTORE 80 :: FOR I=1 TO 96 :: READ A,B,C :: CALL SOUND(200,A,0,B,5,C,5):: 
CALL KEY(0,K,S):: IF S=0 THEN 1310 ELSE IF K=110 OR K=78 THEN 680
1310 IF K=106 OR K=74 THEN 1110
1320 NEXT I :: GOTO 1300 :: GOTO 680
1330 CALL MOTION(#15,0,0,#16,0,0):: CALL HCHAR(24,16,88):: CALL DELSPRITE(#14)
1340 CALL SOUND(-600,262,0):: CALL SOUND(300,330,0):: CALL SOUND(300,392,0):: CA
LL SOUND(600,523,0,392,0,330,0)
1350 CALL SOUND(600,523,0,440,0,349,0):: CALL SOUND(600,523,0,392,0,330,0):: FOR
 I=1 TO 100 :: NEXT I
1360 RESTORE 1380 :: FOR I=18 TO 26 :: READ B :: CALL HCHAR(15,I,B):: NEXT I ::
FOR I=1 TO 500 :: NEXT I :: GOSUB 820
1370 GOTO 80
1380 DATA 71,65,77,69,32,79,86,69,82
1390 SUB CRASH(SP,PP):: CALL MOTION(#14,0,0,#15,0,0,#16,0,0):: CALL SOUND(-1000,
-5,0)
1400 CALL PATTERN(#14,92):: CALL COLOR(#14,10):: FOR I=1 TO 200 :: NEXT I :: CAL
L POSITION(#16,G,H)
1410 FOR I=1 TO PP :: CALL HCHAR(24,SP,88):: SP=SP-1 :: CALL LOCATE(#16,G,H-I*8)
:: NEXT I :: FOR I=1 TO 50 :: NEXT I :: SUBEND
1420 SUB MELODY1(ZU,SP,PT):: FOR I=262 TO 523 STEP 30 :: CALL SOUND(-90,I,5):: C
ALL SOUND(-90,I+30,5):: CALL SOUND(-90,I+60,5):: NEXT I
1430 CALL SOUND(800,523,5,392,5,330,5):: FOR I=1 TO 300 :: NEXT I
1440 CALL POSITION(#15,G,H):: ZE=(22-G+90)*ZU :: DISPLAY AT(18,14)SIZE(8):(22-G+
90)*10;"@";ZU
1450 CALL MOTION(#16,0,1):: FOR I=1 TO ZE :: PT=PT+10
1460 DISPLAY AT(20,20)SIZE(LEN(STR$(PT))+1):PT :: FOR AB=1 TO 7 :: CALL SOUND(-2
00,739,AB):: NEXT AB
1470 IF SP=32 THEN CALL MOTION(#16,0,0):: GOTO 1500
1480 FOR A=1 TO 40 :: NEXT A
1490 IF I/3=INT(I/3)THEN CALL HCHAR(24,SP,127):: SP=SP+1
1500 NEXT I :: CALL MOTION(#16,0,0):: CALL HCHAR(18,14,88,10):: SUBEND
1510 SUB MELODY2 :: FOR I=523 TO 262 STEP -30 :: CALL SOUND(-90,I,5):: CALL SOUN
D(-90,I-30,5):: CALL SOUND(-90,I-60,5):: NEXT I
1520 CALL SOUND(500,131,5,165,5,196,5):: FOR I=1 TO 500 :: NEXT I :: SUBEND
```

```
1530 SUB BILD(PP):: CALL CLEAR :: CALL SCREEN(11):: CALL DELSPRITE(ALL)
1540 DISPLAY AT(2,5):"WAHL DER SPIELSTAERKE:" :: DISPLAY AT(6,5):"1=LEICHT" :: D
ISPLAY AT(8,5):"2=MITTEL"
1550 DISPLAY AT(10,5):"3=SCHWER" :: DISPLAY AT(15,5):"IHRE WAHL?"
1560 ACCEPT AT(15,15)VALIDATE("123")BEEP SIZE(1):PP :: FOR I=1 TO 500 :: NEXT I
:: SUBEND
```



Wheely

ZX-Spectrum 16/48 K

Jagen Sie Ihre Moto-X-Maschine über Stock und Stein!

Ziel des Spieles ist es, eine möglichst große Strecke mit dem Geländemotorrad zurückzulegen. Dies ist gar nicht so einfach, da mehrere Hindernisse zu überwältigen sind. Ein Stein oder eine Mulde darf auf keinen Fall mit dem Vorderrad berührt werden, daher gibt es nur zwei Möglichkeiten: ein Sprung (J) oder auf dem Hinterrad fahren (W). Andernfalls bleibt ein Sturz unausweichlich. Bei einem Sprung sollte man vermeiden, mit einem Vogel zusammenzustoßen. Große Mulden fährt man hinunter, in diesen darf (D) nicht mehr gedrückt werden. Es sind auch Höhlen vorhanden, die man entweder überspringt oder durchquert. Ein Sprungversuch in einer Höhle oder Mulde endet immer mit einem Sturz.

Ab und zu taucht ein zweites, langsameres Motorrad auf, dem man nur durch Überspringen ausweichen kann. Nach einiger Zeit kommt ein Verfolger in Sicht. Dieser ist schneller und ist nur durch übermäßiges Ablassen von Auspuffgasen abzuhängen.

Während der ganzen Fahrt wird Benzin verbraucht, beim Springen, auf einem Rad fahren und Auspuffqualmablassen steigt der Verbrauch gewaltig. Um nicht wegen Spritmangel auf der Strecke zu bleiben, besteht die Möglichkeit, mit den von Zeit zu Zeit auftauchenden Kanistern den Tank aufzufüllen. Dies geschieht durch Berühren derselben.

Die Punktzahl richtet sich nach der Länge der überwundenen Strecke. Der Start erfolgt auf dem Hinterrad, indem man (W) drückt.

Und nun viel Spaß!!!

Programmaufbau:

Zeile

0– 8	Copyright-Zeilen
9– 260	Variablen vorbereiten, Startbild
1000–1410	Hauptprogramm mit Tastaturabfrage und Hinderniskontrolle
2000–2040	DATAs für die Hindernisstrecke
4000–4240	Ausdruck der Hindernisstrecke am rechten Bildschirmrand
6000–6540	Ausdruck bei Kollision mit Hindernis oder Sturz, Spielende
7000–7200	Fremder Motorradfahrer (Verfolger)
9990	Laden des neuen Zeichensatzes, Scroll-MC
9999	Programm abSAVEn

Variablen-Liste:

B = automatisch Bergauffahren bei B = 1
C = Punktzahl
D = Motorrad nach vorn geneigt (D>2), waagerecht (D = 2), Vorderrad oben (D = 1)
E = Grasbüschel bei Bergabfahrt ausdrucken (E = 1)
I = x-Position des Verfolgers
J = Sprung
K = Benzin-Kanister (K = 1)
M = x-Position des Vogels beim Startbild
P = Benzin in 10tel Litern
Q = Auspuffqualm (Q = 1)
R = Verfolger vorhanden (R = 1)
S = Stein im Weg (S = 1)
T = Ausdruck des Steins nach Überfahren
U = Verfolger wird von Auspuffgasen eingequalmt (U = 1)
V = Vogel in der Luft (V = 1)
W = READ-Variable für die Hindernisstrecke
Y = y-Position des Motorrades
N = FOR-NEXT Schleifen
A, F, G, H, L, O, X, Z = Feststehende Werte, um Speicherplatz zu sparen

Tastaturbelegung:

(W) .... Starten/auf einem Rad fahren (wheely)
(D) .... Hinunterfahren (down)
(S) .... Auspuffqualm (smoke)
(J) .... Sprung (jump)
(A) .... neu Starten nach Spielende

```
 150 PRINT "ANFANG DES VERGLEICH
S    :";
 160 GOSUB 1000
 170 PRINT AT 10,5;"START ? =NL"
 180 INPUT A$
 190 PRINT AT 10,5;" ICH SUCHE "
 200 LET Q=USR 16520
 210 PRINT AT 10,5;"ADRESSE BEI
:";Q
 220 LET H=4
 230 LET H$="0000"
 240 LET Z=INT (Q/16)
 250 LET H$(H)=CHR$ (Q-16*Z+28)
 260 LET Q=Z
 270 LET H=H-1
 280 IF Q>0 THEN GOTO 240
 290 PRINT AT 12,5;"HEX ADRESSE
:";H$
 300 PRINT AT 20,6;"NOCHMAL JA/N
EIN ?"
 310 INPUT B$
 320 IF B$(1)<>"J" THEN STOP
 330 CLS
 340 GOTO 100
 350 CLS
 360 PRINT AT 10,5;"ANZAHL DER B
YTES ? :";
 370 INPUT A
 380 PRINT A
 390 FOR N=16620 TO 16620+A-1
 400 SCROLL
 410 PRINT N;"=-->";
 420 INPUT A$
 430 PRINT A$
 440 POKE N,16*(CODE A$(1)-28)+C
ODE A$(2)-28
 450 NEXT N
 460 GOTO 103
1000 INPUT A
1010 PRINT A
1020 LET M=INT (A/256)
1030 POKE B,A-256*M
1040 POKE B+1,M
1050 LET B=B+2
1060 RETURN
1070 STOP
1080 SAVE "BYTESUCHB"
1090 RUN
```

Billard

TI-99/4A

Ein Spiel für Leute, die die Elfenbeinkugel beherrschen

(M.K.) Bei diesem überaus interessanten Spiel ist Ihre Spielstärke, Schnelligkeit, Geschicklichkeit und Ihr Ballgefühl gefragt! Das ist was für Billard-Fans und für solche, die es werden wollen. Holen Sie sich doch einfach „Ihren eigenen Billard-Salon in die gute Stube"!

Nach dem Start des Spiels erscheint zunächst das Titelbild. Sobald das Titelbild fertig aufgebaut ist, erscheint nach dem Drücken einer beliebigen Taste das Spielfeld. Danach wählt man mit dem Joystick ≠ 1 das Loch aus, in das der Spielball gespielt werden soll. Die Auswahl erfolgt über links/rechts. Ist das Loch gewählt, so drückt man auf die Feuertaste. Das Loch wird nun gelb markiert. Es erscheint der Spielball und das Kreuz zur Richtungsabfrage. Nun wählt man mit dem Joystick die Richtung. Anschließend drückt man wiederum auf die Feuertaste, und unten links erscheint eine schwarze Zahl. Jetzt wählt man mit dem Joystick links/rechts die Schußstärke von 1 = langsam; 8 = am schnellsten aus. Ist die Schußstärke gewählt, so drückt man erneut die Feuertaste. Jetzt wird die schwarze Zahl und das Kreuz gelöscht, und der Spielball bewegt sich in gewünschter Richtung und Geschwindigkeit.

Fall 1) Rollt der Ball aus, so muß die neue Richtung und die neue Schußstärke gewählt werden. Der Spielball bleibt dabei auf der Stelle liegen, wo er ausgerollt ist. Die Punktzahl wird dabei nicht erhöht.

Fall 2) Läuft der Ball in das falsche Loch, so wird der Ball wieder an die Anfangsposition gesetzt, und die Richtung und Geschwindigkeit muß neu gewählt werden. Die Punktzahl wird dabei nicht erhöht.

Fall 3) Läuft der Ball in das richtige Loch, wird zunächst geprüft ob mindestens eine Bande berührt wurde. Wenn dies der Fall ist, wird der Punktestand mit 20 multipliziert und man kann wieder das Loch wählen. Wenn keine Bande berührt wurde, geht das Spiel wie in Fall 2) geschildert, weiter.

In allen drei Fällen wird natürlich der Schußzähler um 1 erhöht.

Wenn das Richtungskreuz genau unter den Ball plaziert wird, dann schützt sich das Programm selbständig vor einer Fehlermeldung, wenn man neue Werte eingibt.

Die Bedienung ist, da alle Eingaben über den Joystick 1 erfolgen, sehr einfach. Manchmal kann es aber dazu kommen, daß bei zu großer Geschwindigkeit die Kugel etwas über die Bande herausrollt, da die Abfrage für die Richtungsänderung mit der COINC-Abfrage harmonieren muß. Wenn man die Kugel zu schnell wechseln läßt, kommt die COINC-Abfrage zu kurz, und ein Treffer wird nicht registriert. Dieses tritt jedoch nur bei größeren Geschwindigkeiten auf.

Erläuterungen:

Zeile:

40– 90	Programmkopf
100–170	Datas' für breiten Schriftsatz
180–210	Datas' für Sprites und andere Grafik
220–240	Datas' für den Tisch
250–270	Lesen der Datas' und setzen der Farben
280–300	Titelbild
310–380	Aufbau des Spielfeldes
390–430	Abfrage – in welches Loch gespielt werden soll
440–470	Abfrage – in welche Richtung gespielt werden soll
480–540	Abfrage der Schußstärke
550–650	Schuß mit Bandenabfrage und Lochabfrage
660–690	Abfrage ob richtig eingelocht wurde – mind. 1 Bande
700–730	Subroutine für das Titelbild

Alle „!" können ersatzlos entfallen.

Wem der Normzeichensatz genügt, kann die Zeilen 100–170 weglassen, und die Zeile 260 wie folgt ändern: 260 FOR A=91 TO 123 :: ... Wem der Vorspann (Titelbild) zu lange dauert kann Zeile 280–300, und das Subprogramm Zeile 700–730 weglassen.

Alles andere muß eingetippt werden. Das Programm reduziert sich dann auf 4,6 Kbyt anstatt 5,5 Kbyt.

Variablenliste und Sprites:

Sprite ≠ 1	Spielball
Sprite ≠ 2	Richtungskreuz
Sprite ≠ 3–≠ 8	Löcher
Sprite ≠ 9	Sprite für die Schußstärke
PO	Punkte
SH	Schüsse
I,A	Allg. Laufvariablen
B§	Read Variable
K,S	Key-Rückgabevariable
X,Y	Joyst-Rückgabevariable
SC	Schußstärke (1–8)
BA	Zählvariable für die Bandenberührung mind. 1 Bande um Punkte zu bekommen (für jede Bande 20 Punkte)
LO	Loch (3–8) 3 = links oben; 8 = rechts unten
Z	Distanz zwischen ≠ 1 und ≠ 2
Y2,X2	Position ≠ 2 (für Schußrichtung)
Y1,X1	Position ≠ 1 (für Schußrichtung)
B	Coinc-Abfrage
VX,VY	Geschwindigkeit von ≠ 1
YX,YY	Geschwindigkeitsabnahme von ≠ 1
PY,PX	Positionsabfrage von ≠ 1 (für Richtungsänderung an den Banden)

Routine für Titelbild:

Z	Zeile in der der Text eingeschoben wird
W§	Text
S§	Teil des Textes (Buchstabe) der geschoben wird
Q	Errechnete Spaltenposition für den 1. Buchstaben
AX	Laufvariable für die Anzahl der Buchstaben im Text
AZ	Laufvariable für das Schieben eines Buchstaben

Ein „spitzenmäßiges" Spiel!

```
40 !BILLIARD  5.5 Kbyt
50 !RAINER GAWRIKOW
60 !SOMMERHAUSEN 15
70 !TEL.:02245/3983
80 !5203 MUCH
90 !
100 !SCHRIFT 33-90
110 DATA 1818181818180018,6C6C6C,6CFEFE6C6CFEFE6C,6CFED0FC7E16FE6C,C6CE1E3C78F0E
6C6,70F8D0767ECCFE76,18182,1818303030301818,3030181818183 03
120 DATA 0000001818,0018187E7E1818,0000000000018182,0000007E7E,0000000000001818,0
60E1E3C78F0E0C,7CFECEDEF6E6FE7C,1838787818181818,7CFE0E3C78E0FEFE
130 DATA 7CFE0E3C3C0EFE7C,1C3C7CECFEFE0C0C,FEFEC0FCFE06FE7C,7CFEC0FCFEC6FE7C,FEF
E060E1C3838303,7CFEC67C7CC6FE7C,7CFEC6FE7E06FE7C,00181800001818
140 DATA 001818000018182,061E7CF0F07C1E06,007C7C00007C7C,C0F07C1E1E7CF0C,7CFE061
E1C001818,7CFEC6DEDEDEDEC,387CEEC6FEFEC6C6,FCFEC6FCFCC6FEFC
150 DATA 7CFEC0C0C0C0FE7C,FCFEC6C6C6C6FEFC,FEFEC0F8F8C0FEFE,FEFEC0F8F8C0C0C0,7CF
EC0CECEC6FE7C,C6C6C6FEFEC6C6C6,1818181818181818,FEFE06060606FE7C
160 DATA C6CEDCF8F8DCCEC6,C0C0C0C0C0C0FEFE,C6EEFEFED6C6C6C6,C6E6E6F6DECECEC6,7CF
EC6C6C6C6FE7C,FCFEC6FEFCC0C0C0,FCFEC6C6D6CCFE76,FCFEC6FEFCDCCEC6
170 DATA 7CFEC0FC7E06FE7C,FEFE181818181818,C6C6C6C6C6C6FE7C,C6C6EE6C7C38381,C6C6
C6C6D6FEFE6C,C6EE7C38387CEEC6,C6CEFC783030303,FEFE0E1C3870FEFE
180 !DATEN FUER GRAFIK 91-111
190 DATA 3C4299A1A199423C,3C7EFFFFFFFF7E3C,000010107C101,,
200 DATA 0000000000000FF,0000000000000FFFF,0000000000FFFFFF,00000000FFFFFFFF,000
000FFFFFFFFFF,0000FFFFFFFFFFFF,00FFFFFFFFFFFFFF,FFFFFFFFFFFFFFFF
210 DATA FFFFFFFFFF7E3C,003C7EFFFFFFFFFF,387CFEFFFF7E3C18,1C3E7FFFFF7E3C18,183C7
EFFFFFE7C38,183C7EFFFF7F3E1C,0000001818,
220 !DATEN TISCH 112-124
230 DATA FFFFFFFFFFFFFFFF,F0F0F0F0F0F0F0F,0F0F0F0F0F0F0F0F,00000000FFFFFFFF,FFFF
FFFF,,,,1F3F7FFFFFFFFFFF,F8FCFEFFFFFFFFFF,FFFFFFFFFF7F3F1F
240 DATA FFFFFFFFFFFEFCF8
250 CALL CLEAR :: CALL SCREEN(6):: PO,SH=0
260 FOR A=33 TO 123 :: READ B$ :: CALL CHAR(A,B$):: NEXT A
270 FOR A=1 TO 8 :: CALL COLOR(A,16,1):: NEXT A :: CALL COLOR(10,16,4,11,7,13,12
,7,1)
280 !TITELBILD
290 CALL STEXT(8,"R.G. SOFT PRESENTS"):: CALL STEXT(12,">>BILLARD<<"):: CALL STE
XT(16,"© 85' BY R.GAWRIKOW"):: CALL STEXT(24,"PRESS ANY KEY TO START !")
300 CALL KEY(3,K,S):: IF S=0 THEN 300
310 !SPIELFLAECHE
320 CALL CLEAR :: CALL HCHAR(3,1,111,480):: CALL HCHAR(1,1,120):: CALL HCHAR(1,2
,112,30):: CALL HCHAR(1,32,121):: CALL HCHAR(2,2,112)
330 CALL HCHAR(2,3,116,28):: CALL HCHAR(2,31,112):: CALL VCHAR(2,1,112,17):: CAL
L VCHAR(2,32,112,17):: CALL VCHAR(3,2,113,15)
340 CALL VCHAR(3,31,114,15):: CALL HCHAR(18,2,112):: CALL HCHAR(18,3,115,28):: C
ALL HCHAR(18,31,112):: CALL HCHAR(19,1,122)
350 CALL HCHAR(19,2,112,30):: CALL HCHAR(19,32,123):: CALL HCHAR(10,9,110):: CAL
L HCHAR(10,24,110)
360 CALL SPRITE(#3,106,2,12,12,#4,105,2,9,128,#5,107,2,12,238)
370 CALL SPRITE(#6,108,2,133,12,#7,104,2,137,128,#8,109,2,133,237)
380 DISPLAY AT(21,1)BEEP:"12345678   SHOOTS:   POINTS:" :: DISPLAY AT(23,1):"\ab
```

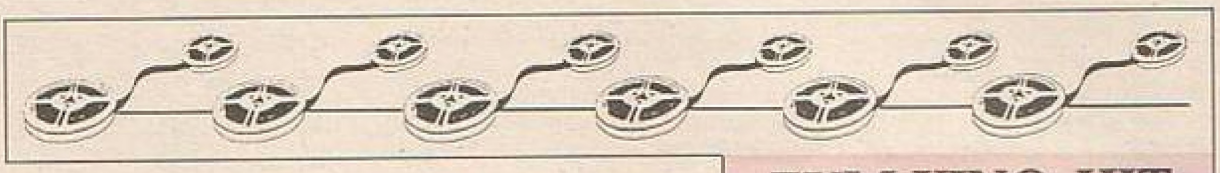
```
cdef9   000000    000000"
390 !LOCHABFRAGE
400 LO=3
410 FOR I=1 TO 50 :: NEXT I :: CALL JOYST(1,X,Y):: CALL COLOR(#LO,2):: LO=LO+X/4
420 IF LO>8 THEN LO=3 ELSE IF LO<3 THEN LO=8
430 CALL COLOR(#LO,11):: CALL KEY(1,K,S):: IF K<>18 THEN 410
440 !RICHTUNGSABFRAGE
450 CALL SPRITE(#1,92,7,73,184)
460 BA=0 :: CALL SPRITE(#2,93,2,73,128):: FOR I=1 TO 100 :: NEXT I
470 CALL JOYST(1,X,Y):: CALL MOTION(#2,-Y*2,X*2):: CALL KEY(1,K,S):: IF K<>18 TH
EN 470 ELSE CALL MOTION(#2,0,0)
480 !SCHUSSSTAERKE
490 SC=1
500 CALL JOYST(1,X,Y):: SC=SC+X/4
510 IF SC<1 THEN SC=8 ELSE IF SC>8 THEN SC=1
520 CALL SPRITE(#9,SC+48,2,161,9+SC*8):: FOR I=1 TO 50 :: NEXT I
530 CALL KEY(1,K,S):: IF K<>18 THEN 500
540 CALL DELSPRITE(#9)
550 !SCHUSS
560 CALL POSITION(#2,Y2,X2,#1,Y1,X1):: SH=SH+1 :: DISPLAY AT(23,12)SIZE(6)BEEP:U
SING "######":SH
570 CALL DISTANCE(#1,#2,Z):: VX=(-X1+X2)/SQR(Z)*5*SC :: VY=(-Y1+Y2)/SQR(Z)*5*SC
:: IF ABS(VX)>100 OR ABS(VY)>100 THEN 460
580 CALL DELSPRITE(#2):: YX=ABS(VX/40):: YY=ABS(VY/40)
590 CALL MOTION(#1,VY,VX):: CALL COINC(ALL,B):: IF B=-1 THEN 670
600 CALL COINC(#1,#LO,8,B):: IF B=-1 THEN 670
610 CALL POSITION(#1,PY,PX):: IF (VY>0 AND PY>127-ABS(VY/2))OR(VY<0 AND PY<17+AB
S(VY/2))THEN VY=-VY :: BA=BA+1 :: CALL SOUND(-20,110,0)
620 CALL COINC(ALL,B):: IF B=-1 THEN 670 ELSE CALL MOTION(#1,VY,VX)
630 IF (VX>0 AND PX>232-ABS(VX/2))OR(VX<0 AND PX<17+ABS(VX/2))THEN VX=-VX :: BA=
BA+1 :: CALL SOUND(-20,110,0)
640 CALL COINC(ALL,B):: IF B=-1 THEN 670 ELSE CALL MOTION(#1,VY,VX)
650 VY=VY-YY*SGN(VY):: VX=VX-YX*SGN(VX):: IF INT(ABS(VY))=0 AND INT(ABS(VX))=0 T
HEN CALL MOTION(#1,0,0):: GOTO 460 ELSE 600
660 !EINGELOCHT ?
670 CALL MOTION(#1,0,0):: CALL COINC(#1,#LO,8,B):: IF B<>-1 THEN CALL SOUND(-100
,110,0):: GOTO 450 ELSE CALL DELSPRITE(#1)
680 CALL SOUND(-100,1000,0):: IF BA>=1 THEN PO=PO+BA*20 :: DISPLAY AT(23,22)SIZE
(6)BEEP:USING "######":PO ELSE CALL SOUND(-100,110,0):: GOTO 450
690 CALL COLOR(#LO,2):: GOTO 400
700 !ROUTINE FUER TITELBILD
710 SUB STEXT(Z,W$):: IF LEN(W$)>30 THEN SUBEXIT
720 Q=INT(32-LEN(W$))/2+1 :: FOR AX=Q TO LEN(W$)+Q-1 :: S$=SEG$(W$,AX-Q+1,1):: F
OR AZ=31 TO AX STEP -1
730 CALL HCHAR(Z,AZ+1,32):: CALL HCHAR(Z,AZ,ASC(S$)):: NEXT AZ :: NEXT AX :: SUB
END
```

Restabdruck: „SOLARIS" (MSX)

```
2550 DATA T120S8M2000003FAAFAAB-04CC03B-04CC03B-04CC03B-04CC03FAAFAAFAAA04C#C#DF
F03GBBFAAB-04CC03FAAFAA
2560 DATA 17,93,0,2,17,98,-1,4,16,108,-5,12,16,112,-7,18,15,116,-9,21,15,122,-11
,25,14,127,-13,29,14,135,-15,33,13,142,-17,37,13,153,-17,37,12,163,-20,39,12,176
,-20,39
2570 DATA 11,11,10,10,9,9,8,8,7,7,6,6
2580 DATA 0207070D1F377FDB,FDBB3F67C3E7FC70,40E0E0B0F8ECFEDB,BFDDFCE6C3E73F0E,00
00020F1F3F7FFF,FFBF3F7F7F7C7000,000040F0F8FCFEFF,FFFDFCFEFE3E0E00
2590 DATA 0103071E3FF7FB37,3F67C3E7FF710000,40E0E0F0B8F8FCEE,DFECFCE6C3E73F0E,00
00010F3FFFFF3F,3F7FFFFF71000000,000040F0F8F8FCFE,FFFCFCFEFE7E0C00
2600 DATA 0207070F1D1F3F77,FB373F67C3E7FC70,80C0E078FCEFDFEC,FCE6C3E7FF8E0000,00
00020F1F1F3F7F,FF3F3F7F7F7E3000,000080F0FCFFFFFC,FCFEFFFF8E000000
2610 DATA 03071C317FFC613F,6DD6FFFF83F87F07,C0E0300CFE3F86FC,B66BFFFFC11FFEE0,00
01070D1F3E190F,1F363F3F331E0700,0000E0B0F87C98F0,F86CFCFCCC78E000
2620 DATA 00000103070F0503,070F0F0F05030000,000080C0E0F0A0C0,E0F0F0F0A0C00000,00
00000001030101,0307030100000000,0000000080C08080,C0E0C08080000000
2630 DATA 0000000000000103,0100000000000000,0000000000000C0,0000000000000000,00
00000000000001,0000000000000000,0000000000000000,0000000000000000
```

TRONIC PROUDLY PRESENTS:

Ein Super-Bond-Abenteuer für den TI-99!



ZUM KINO-HIT

James Bond 007 Im Angesicht des Todes

Sie kennen den Film? Dann also los, schlüpfen Sie in die Rolle von 007: Retten Sie Stacey vor Zorin's Leuten!

James Bond schlägt momentan alle (Kino)-Rekorde. Der Film „A View To A Kill" (Im Angesicht des Todes) entpuppt sich – wie könnte es anders sein? – als Kassenschlager. Auch die Titelmusik von Duran Duran wurde millionenfach verkauft. Wen wundert's, daß es das neueste Bond-Abenteuer auch schon als Software-Game gibt (von DOMARK)? HOMECOMPUTER bietet seinen Lesern – und TI-99-Besitzern – nun ein kleines Bonbon an: „Im Angesicht des Todes" auch bei uns „erhältlich"!

Bei diesem wahrlich hervorragenden Programm geht es darum, sich in die Rolle des Geheimagenten Seiner Majestät, James Bond 007, hineinzuversetzen. Sie befinden sich in der unterirdischen Mine des Edel-Schurken Max Zorin, der sich auf sein Projekt „Mainstrike" vorbereitet. Sie – als 007 – müssen nun die schöne und „unschuldige" Stacey Sutton retten. Dies gelingt Ihnen nur, wenn Sie dem Bösewicht Zorin ausweichen und nacheinander – in der richtigen Reihenfolge – die Buchstaben J, A, M, E, S, B, O, N, D berühren.

Doch: Rechnen Sie mit der Hinterlist des Schurken Zorin! Er hat überall in der Mine Sprengladungen angebracht (magenta-rot dargestellt), um durch eine Explosion ein Erdbeben hervorzurufen, das ganz Sillicon Valley dem Erdboden gleichmachen würde.

Wenn James Bond einen Teil des Sprengstoffes berührt, kann er ein Stück des Stollens wegsprengen, indem er den Joystick-Hebel nach unten zieht. Durch Druck auf die Feuertaste kann 007 auch springen.

Desweiteren sind im Stollen noch zwei „Q" verteilt. Zorin kann diese Hindernisse nicht überwinden. Falls Sie ein „Q" berühren sollten, und den Joystick-Hebel nach unten ziehen, so werden die Figuren wieder in die Ausgangsstellung gebracht. Das funktioniert jeweils nur einmal. Achtung: Sollten Sie in das angestaute Wasser fallen, haben Sie und Roger Moore Ihr Leben ausgehaucht.

„Gestatten, mein Name ist Bond – James Bond! Ich sorge dafür, daß Sie knisternde Spannung vermittelt bekommen."

Ein Taktik-Spiel der Spitzenklasse!

Nach der Eingabe des Programms und Starten mit RUN erscheint das „Orignal-James Bond-Titelbild". Das Spiel ist mit einer Anleitung versehen und bietet dem Bond-Fan zwei Schwierigkeits-Stufen an. (M.K.)

TI-99/4A – Extended Basic

„AUF GEHT'S, 007!

```
100 ! James Bond oo7                Im Angesicht Des Todes      A View To A Kill
          (c)1985 Patrick Schmitz       Am Zehnthof 4
110 ! D-5480 Rem.-Unkelbach         Germany                     Version 1.o/Ext.Ba
sic        on TI-99/4A Homecomp.
120 ! Albert R.Broccoli             presents ROGER MOORE        as Ian Fleming's
          JAMES BOND 007
130 ! Music by John Barry           Story by R.Maibaum          and Mike G.Wilson
          progr.by Pat.W.Schmitz        all rights reserved
140 ON BREAK NEXT :: OPTION BASE 1 :: CALL CLEAR :: CALL SCREEN(12):: DIM P(80):
: RESTORE 330 :: FOR Z=1 TO 8 :: READ A$ :: CALL CHARPAT(ASC(A$),A$):: CALL CHAR
(39+Z,A$):: NEXT Z
150 CALL CHAR(49,"007824242424247800 61FF6060C0000000008"):: E=6 :: R=12 :: GOSUB
 970 :: DISPLAY AT(21,7):"( )*+, -./1 00723" :: GOSUB 760 :: CALL DELSPRITE(ALL):
: CALL CLEAR
160 CALL CHARSET :: FOR Z=65 TO 90 :: CALL CHARPAT(Z,A$):: A$="0000"&SEG$(A$,1,4
)&SEG$(A$,7,4)&SEG$(A$,13,4):: CALL CHAR(Z+32,A$):: NEXT Z :: CALL CHAR(115,"000
0003C40380478")
170 CALL CHAR(99,"0000003C4040403C",106,"0000000808084830000000242830282 4"):: DI
SPLAY AT(1,1)ERASE ALL:" - IM ANGESICHT DES TODES -": :"    - a view to a kill -
"
180 DISPLAY AT(12,1):"-1- Einfuehrung": :"-2- Spielanleitung": :"-3- Spiel - ein
fach": :"-4- Spiel - fuer Bond-Fans": : : :"Bitte Taste '1','2','3' oder": :"'4'
 druecken."
190 CALL KEY(0,X,Y):: IF Y=0 OR X<49 OR X>52 THEN 190 ELSE CALL SOUND(-50,1400,5
,1402,7):: CALL CLEAR :: IF X=49 THEN 200 ELSE IF X=50 THEN 260 ELSE IF X=51 THE
N P(3)=2 :: GOTO 290 ELSE P(3)=-1 :: GOTO 290
200 DISPLAY AT(1,1):"Roger Moore ist James Bond  oo7: 'Mein Name ist Bond,   Jam
es Bond.'                Er hat die Lizens zu toeten."
210 DISPLAY AT(6,1):"Fuer ihn heisst es 'Leben   und sterben lassen'. Er     bes
iegt Killer wie den 'Mannmit dem goldenen Colt'."
220 DISPLAY AT(10,1):"Fuer viele 'Bondinen' ist er'Der Spion,der mich liebte'.Se
in bester Auftrag  ist    'Moonraker' - streng geheim,"
230 DISPLAY AT(14,1):"versteht sich.              Doch James Bond ist immer   wi
eder 'In toedlicher       Mission' unterwegs, manchmalauch mit 'Octopussy'."
240 DISPLAY AT(20,1):"Laechelnd und Wodka-Martini schluerfend (geschuettelt,  ni
cht geruehrt) steht James Bond oo7 immer wieder"
250 DISPLAY AT(24,1):"'Im Angesicht des Todes'..." :: CALL KEY(0,X,Y):: IF Y<>1
THEN 250 ELSE CALL SOUND(-100,1400,5,1402,7):: GOTO 170
260 DISPLAY AT(6,1):"007 muss die Buchstaben J,A,M,E,S,B,O,N & D beruehren.  Zor
in versucht ihn daran zu hindern. Beruehrt man den   Sprengstoff, so wird ein"
270 DISPLAY AT(11,1):"Stueck des Stollens weggge-  sprengt, wenn man den Joy-  st
ickhebel nach unten zieht.Auf diese Weise kann man    auch die Figuren in Ihre"
280 DISPLAY AT(16,1):"alte Position bringen, wenn man ein 'Q' beruehrt.":"ein Fa
ll in das Wasser ist  immer toedlich." :: GOTO 250
290 CALL CLEAR :: GOSUB 720 :: CALL CHAR(80,"784444447850484444 6C545454444444",8
5,"0404040404044438782424382424247 8"):: CALL COLOR(14,8,2)
300 CALL CHAR(87,"3C1818181818183C3C66667E66666666FC6666666666666FCFF181818181818
18"):: CALL CHAR(60,"00000000FF000000000000FF00FF0000"):: FOR Z=1 TO 8
310 CALL COLOR(Z,8,2):: NEXT Z :: DISPLAY AT(3,1):" < WM XNGESICHT YES ZODES <"
:: IF P(3)=-1 THEN DISPLAY AT(1,1):"POGER QOORE = UAMES VOND 007" ELSE DISPLAY A
T(1,8):"NOCH 3 CHANCEN"
320 CALL CHAR(136,"000070FF5070C0C0000020F",138,"00000EFF0A0E03030000060F"):: CA
LL HCHAR(2,2,136):: CALL HCHAR(2,3,137):: CALL HCHAR(2,30,139)
330 DATA J,A,M,E,S,B,O,N,D,247,10,220,20,196,7,175,7,196,25
340 CALL CHAR(140,"0000FF"):: CALL HCHAR(2,31,138):: M=P(3):: RESTORE 330 :: FOR
 Z=1 TO 9 :: READ R$(Z):: NEXT Z :: X=0 :: FOR Z=1 TO 9 :: RANDOMIZE
350 E,P(Z)=INT(RND*14)+5 :: R=INT(RND*26)+4 :: CALL GCHAR(E,R,A):: CALL GCHAR(E+
1,R,T):: IF A<>32 OR T<>102 THEN 350
360 FOR T=1 TO Z-1 :: IF P(T)=E THEN X=X+1 :: IF X=2 THEN X=0 :: GOTO 350
370 NEXT T :: CALL HCHAR(E,R,ASC(R$(Z))):: CALL SPRITE(#10+Z,ASC(R$(Z)),10,E*8-6
,R*8-7):: NEXT Z :: P(1)=1 :: CALL CHAR(40,"060C081C3E7F3E1C150240AA04205"):: FO
R Z=1 TO 6
380 E=INT(RND*14)+5 :: R=INT(RND*26)+4 :: CALL GCHAR(E,R,A):: CALL GCHAR(E+1,R,T
):: IF A<>32 OR T<>102 THEN 380
390 CALL COLOR(2,14,2):: CALL HCHAR(E,R,40):: NEXT Z :: P(3)=M :: CALL CHAR(123,
"3844444444544834"):: FOR Z=1 TO 2
400 E=INT(RND*10)+10 :: R=INT(RND*26)+4 :: CALL GCHAR(E,R,A):: CALL GCHAR(E+1,R,
T):: IF A<>32 OR T<>102 THEN 400
410 CALL HCHAR(E,R,123):: NEXT Z :: P(2)=0 :: CALL HCHAR(2,4,140,26):: CALL SOUN
D(-99,880,3,881,4):: CALL HCHAR(2,4,32,26)
420 CALL CHAR(98,"1C7F49497F49633E",103,"1C7F49497F63493E"):: CALL SPRITE(#1,98,
12,153,217,#2,103,15,41,49):: E=20 :: R=28 :: M=6 :: S=7
```

```
430 X=(E<M)*2+1 :: Y=(R<S)*2+1 :: CALL GCHAR(M+X,S+Y,A):: IF A>96 AND A<>104 THE
N 440 ELSE M=M+X :: S=S+Y :: CALL LOCATE(#2,M*8-7,S*8-7):: GOTO 450
440 CALL GCHAR(M+X,S,A):: IF A<96 OR A=104 THEN M=M+X :: CALL LOCATE(#2,M*8-7,S*
8-7)ELSE CALL GCHAR(M,S+Y,A):: IF A<96 OR A=104 THEN S=S+Y :: CALL LOCATE(#2,M*8
-7,S*8-7)
450 IF M<>E OR S<>R THEN 470 ELSE CALL SOUND(-999,110,0):: CALL SCREEN(16):: CAL
L SCREEN(7):: CALL SCREEN(2):: CALL SOUND(-999,110,4,111,9):: CALL PATTERN(#1,40
)
460 CALL COLOR(#1,7):: CALL LOCATE(#2,M*8-14,S*8-4):: CALL SOUND(-999,110,7,111,
12):: CALL DELSPRITE(#1):: GOTO 630
470 IF E<21 THEN 500 ELSE CALL PATTERN(#1,41):: CALL LOCATE(#1,E*8-12,R*8-7):: C
ALL SOUND(-999,-6,0):: CALL LOCATE(#1,E*8-9,R*8-7):: CALL COLOR(#1,7)
480 CALL COLOR(12,5,2,13,3,5):: CALL COLOR(#1,12):: CALL COLOR(12,6,2,13,3,6):: 
CALL CHAR(120,"0C3FFFFFFFFFFFFF")
490 CALL SOUND(-444,-7,5):: CALL CHAR(120,"3C7EFFFFFFFFFFFF"):: CALL DELSPRITE(#
1):: GOTO 630
500 CALL KEY(2,X,Y):: IF X<>18 THEN 520 ELSE CALL GCHAR(E-1,R,A):: IF A<>32 THEN
 CALL SOUND(-100,110,4,112,7):: GOTO 520
510 E=E-1 :: CALL SOUND(-200,880-E*18,4,880-E*18,6):: CALL LOCATE(#1,E*8-7,R*8-7
):: GOTO 500
520 CALL JOYST(2,X,Y):: X=SGN(X):: Y=-SGN(Y):: CALL GCHAR(E,R+X,A):: IF A>96 AND
 A<103 OR A=115 OR A=114 THEN CALL SOUND(-9,110,6):: GOTO 550 ELSE R=R+X :: CALL
 LOCATE(#1,E*8-7,R*8-7)
530 IF Y=0 THEN 550 ELSE CALL GCHAR(E+Y,R,A):: IF A<>104 THEN 540 ELSE E=E+Y :: 
CALL LOCATE(#1,E*8-7,R*8-7):: GOTO 550
540 CALL GCHAR(E+Y+1,R,T):: IF T=104 THEN E=E+Y :: CALL LOCATE(#1,E*8-7,R*8-7)EL
SE CALL SOUND(-9,110,6)
550 CALL GCHAR(E+1,R,T):: IF T<>102 AND T<>104 AND T<>128 THEN E=E+1 :: CALL SOU
ND(-100,880-E*18,4):: CALL LOCATE(#1,E*8-7,R*8-7):: GOTO 550 ELSE CALL GCHAR(E,R
,A):: IF A=32 THEN 430
560 IF A<>ASC(R$(P(1)))THEN 590 ELSE P(1)=P(1)+1 :: P(2)=P(2)+P(1)*100
570 FOR T=1 TO ABS(P(1)=10)*2+1 :: FOR Z=220 TO 440 STEP 20 :: CALL SOUND(-10,Z,
6,Z+2,10):: NEXT Z :: CALL HCHAR(E,R,32):: CALL DELSPRITE(#9+P(1)):: NEXT T :: I
F P(1)<10 THEN 430
580 CALL DELSPRITE(ALL):: CALL CLEAR :: CALL CHAR(89,"004444281010101"):: DISPLA
Y AT(1,1):"STACEY SUTTON IST GERETTET.": : :"JAMES BOND:": :"KEINER KANN ES BESS
ER." :: GOTO 660
590 IF A<>40 THEN 610 ELSE CALL JOYST(2,X,Y):: IF Y<>-4 THEN 430 ELSE CALL SOUND
(-200,-6,2):: CALL VCHAR(E,R,32,2):: CALL SPRITE(#10,41,16,E*8+1,R*8-7)
600 CALL SOUND(-99,-5,5):: CALL SOUND(-99,-6,2):: CALL SCREEN(14):: CALL SCREEN(
2):: CALL SOUND(-100,-5,4):: CALL DELSPRITE(#10):: GOTO 550
610 IF A<>123 THEN 430 ELSE CALL JOYST(2,X,Y):: IF Y<>-4 THEN 430 ELSE CALL SCRE
EN(12):: CALL SCREEN(2):: CALL SOUND(-100,880,4,882,8):: CALL SOUND(-100,990,7,9
92,12)
620 CALL HCHAR(E,R,32):: CALL SOUND(-99,1010,9,1012,9):: E=20 :: R=28 :: M=6 :: 
S=7 :: CALL LOCATE(#1,E*8-7,R*8-7,#2,M*8-7,S*8-7):: GOTO 430
630 CALL LOCATE(#2,M*8-7,S*8-7):: IF P(3)>-1 THEN P(3)=P(3)-1 :: DISPLAY AT(1,12
)SIZE(2):P(3)+1 :: IF P(3)=0 THEN DISPLAY AT(1,21):" " :: GOTO 420 ELSE DISPLAY
AT(1,21):"N" :: GOTO 420
640 FOR T=1 TO 2 :: RESTORE 330 :: READ A$,A$,A$,A$,A$,A$,A$,A$,A$ :: FOR Z=1 TO
 5 :: READ E,M :: FOR A=1 TO M/1.7 :: CALL SOUND(-333,110,30,110,30,E*4,30,-4,A)
:: NEXT A :: NEXT Z :: NEXT T
650 CALL CHAR(80,"0078444478404040",85,"004444444444438"):: CALL DELSPRITE(ALL)
:: CALL CLEAR :: CALL MAGNIFY(3):: CALL SPRITE(#1,112,16,33,25,16,0)
660 CALL CHAR(80,"0078444478404040",85,"004444444444438"):: DISPLAY AT(10,6):"1
 - NEUES SPIEL": : :TAB(6);"2 - ENDE" :: DISPLAY AT(24,6):P(2);"PUNKTE"
670 CALL KEY(0,X,Y):: IF X=49 THEN CALL DELSPRITE(ALL):: CALL CLEAR :: CALL SCRE
EN(12):: CALL COLOR(1,2,12):: CALL MAGNIFY(1):: RUN 160 ELSE IF X=50 THEN CALL C
LEAR :: PRINT "007 WILL RETURN": : :: END ELSE 670
680 DATA ceeeeeeeeeeeeeeeeeeeeeeeeecaPr                    da9s          f
ff ffhffffhfccffffffhffffhf      h    h da     h     h     h    h d
690 DATA cffffffffffffhffff ffffffhfca          h          h da    ff ff ffff
ffffffffffhfca  ff                    h dcff  fffff ffhffffffffffffhfc
700 DATA a            h            h dcfffhffffffffffffhfff fffffca   h
  h        da   h ffff        h     hf  dcf fhf    ffff  ffff ffh ffc
710 DATA a   h                h   da  ffffffff    fff  ff h   dcff
      fffffccxxxxxxxxxxxxxxxxxxxxxxxxxxccyyyyyyyyyyyyyyyyyyyyyyyyyyyyc
720 CALL CHAR(112,"C0E03F18362E2D213110171B1837E0C00307FC186C74B4848C08E8D818EC0
703")
730 CALL CHAR(97,"FFFEFFFCFEFFFEFF",99,"FFFFFFFFFFFFFFFFFF7FFF3F7FFFFFFF",101,"0
0FFFFFFFFADA58452DAFFFFFFF5A584",104,"C3FFC3FFC3FFC3FF")
740 CALL CHAR(128,"ABEBFFFFFFFFFFFF",120,"3C7EFFFFFFFFFFFFFFFFFFFFFFFFFF"):: C
```

```
ALL SCREEN(2):: RESTORE 680 :: FOR Z=1 TO 4 :: READ A$ :: PRINT A$ :: NEXT Z
750 DISPLAY AT(24,1):"c"&RPT$(CHR$(128),26)&"c" :: CALL COLOR(9,3,2,10,9,2,11,16
,2,12,6,2,13,3,6):: RETURN
760 RESTORE 770 :: FOR Z=1 TO 91 :: READ A :: FOR E=1 TO 8 STEP .33/(A-INT(A)):: 
 CALL SOUND(-333,A/1.19,30,A/1.19,30,A*10,30,-4,E):: NEXT E :: NEXT Z :: RETURN
770 DATA 131.12,147.12,165.12,147.2,131.12,147.12,165.12,147.35,131.08,165.03,16
5.03,165.03,165.09,147.03,147.03,147.03,147.06
780 DATA 175.03,175.03,175.03,175.09,165.06,165.07,165.08,131.06,165.03,165.03,1
65.03,165.09,147.03,147.03,147.03,147.06,175.03
790 DATA 175.03,175.03,175.11,165.07,147.07,131.08,247.09,220.19,196.06,175.04,1
96.21,165.07,147.07,131.08,247.09,220.19,196.06
800 DATA 175.04,196.4,131.06,147.11,247.07,220.19,165.06,175.05,196.22,175.05,16
5.04,175.16,131.05,165.05,147.3,131.06,147.11
810 DATA 247.07,220.19,165.06,175.05,196.22,175.05,165.04,175.16,131.05,165.05,1
47.55,131.05,147.08,220.06,196.22,131.05,147.08
820 DATA 247.06,220.22,131.05,147.08,262.06,247.65
830 DATA 0000000000000000,000000030F3F7FFF,031F7FFFFFFFFFFF,C0FCFFFFFFFFFFFF,00C
0F4FFFFFFFFFF,00000080E0F0F87C,0000000000000000,0000000000000000
840 DATA 0001010303030707,FFFFFFFFFFFFFFFF,FDFFFFFFFFFFFDFB,FDFBDFFFBFFBFFDF,7FF
FFFFFFFFFFFFF,FCFEFEFFFFFFFFFF,0000808080C0C0C0,0000000000000000
850 DATA 0F0F1F1F1F1F0F1F,FFFFFFFFFFFFF0C0,F7FEE7FDF7D00000,BFF79FFFB8000000,FFF
EF00000000000,FF1F1F0F2F0F1F0F,E0E0E0E0E0F0F8F8,0000000000000000
860 DATA 1F1F3F1F3F3F3F7F,8080C08080C080C0,0000000000000000,0000000000000000,000
0000000000000,0303110323030503,F8F8FCFCFEFEFCFC,0000000000000000
870 DATA 7F7F7F7F3F3F170F,80889D1700030C11,00A0D8FC06E2D8C4,0000010304020 60C,005
0FEAB0138CE9D,03071387EB431301,FCFEFCFEFCFCFCFC,0000000000000000
880 DATA 1F0F1F0F0F0F0F07,1201080000000000,A0C0000100000200,1810181030283018,2A1
C000001000000,6B234B4103030707,FEFCFCFCFCFCF8F8,0000000000000000
890 DATA 0F0F07070701 0000,0000C090C0D0E0F0,0000040C0B010000,18141A1A3AFAF440,004
0000000010002,070F0F170F1F1F3F,FCF8F0F8E0E0C000,0000000000000000
900 DATA 0000000000000000,FFFFFFFF7F7F3F3F,000000011E370000,00000010BDEF0000,000
1030302870606,FFFFFEFEFEFEFCFC,0000000000000000,0000000000000000
910 DATA 0000000000000000,0F0F070707070303,0002000000000C0E0,0008000000000000,050
50D0D0B1B0F1F,FCFCF8F8F0F0E0E0,0000000000000000,0000000000000000
920 DATA 0000000000000000,0000000000000000,FF7F7F3F1F0F0300,FFFFFFFFFFFFEF1C,FFF
EFEFCFCF0C080,C080000000000000,0000000000000000,0000000000000000
930 DATA 64,65,66,67,68,69,70,71,72,73,80,81,82,74,75,56,76,83,84,85,86,87,57,77
940 DATA 78,88,89,90,91,92,79,96,97,93,94,95,104,105,98,99,100,106,107,108,109,1
10,101,102
950 DATA 103,111,112,113,114,115,120,121,122,123,116,117,118,124,125,126,127,128
,119,136,137,129,130,131
960 DATA 131,131,132,133,134,135,131,131
970 CALL CLEAR :: GOSUB 1020 :: RESTORE 930 :: FOR Z=1 TO 80 :: READ P(Z):: NEXT
 Z :: RESTORE 830 :: T=0 :: FOR A=1 TO 10 :: FOR M=1 TO 8 :: T=T+1 :: READ A$ :: 
 CALL CHAR(P(T),A$):: NEXT M :: NEXT A
980 CALL COLOR(4,2,4,5,2,4,6,2,4,7,2,15,8,2,15,9,2,4,10,2,15,11,2,15,12,2,4,13,2
,4,14,2,15)
990 T=0 :: FOR A=E TO E+9 :: IF A=E+7 THEN CALL COLOR(#1,15,#2,15,#3,15,#4,15)EL
SE IF A=E+9 THEN CALL COLOR(#5,15,#6,15,#7,15)
1000 FOR S=R TO R+7 :: T=T+1 :: CALL HCHAR(A,S,P(T)):: NEXT S :: NEXT A :: RETUR
N
1010 DATA 0000041000102000,171F17370B0B0B03,0000080818102000,0501010100000000,1F
0F0F0300000000,FFFFDC0000000000,8000000000000000
1020 RESTORE 1010 :: FOR Z=1 TO 7 :: READ A$ :: CALL CHAR(32+Z,A$):: NEXT Z
1030 DATA 1,5,8,2,8,6,9,2,10,3,10,4,10,5
1040 RESTORE 1030 :: FOR Z=1 TO 7 :: READ P(Z),R$(Z):: NEXT Z :: FOR Z=1 TO 7 :: 
 CALL SPRITE(#Z,32+Z,1,(E-1+P(Z))*8-7,(R-1+VAL(R$(Z)))*8-7):: NEXT Z :: RETURN
```

Rein in's Abenteuer! – JAMES BOND 007 läßt grüßen

SINCLAIR-Refinanzierung nicht mehr erforderlich

Aufgrund der großen Verkaufserfolge in jüngster Zeit ist die vorgeschlagene Refinanzierung von Sinclair Research Ltd. nicht mehr erforderlich. Dies gab Sir Clive Sinclair in London bekannt. Wörtlich: „Unser Anteil am Markt in Großbritannien ist in den vergangenen Monaten beständig auf gesunde 40 Prozent angestiegen, was zweifelsohne unsere Marktführerschaft bei Home Computern bestätigt." In den vergangenen Tagen sei ein sehr vielversprechender Vertrag mit der in Großbritannien ansässigen Dixons-Gruppe in Höhe von über 10 Millionen Pfund Sterling abgeschlossen worden, zahlbar an Sinclair innerhalb der kommenden drei Monate.

„Wir können jetzt der Zukunft vertrauensvoll entgegenblicken" erklärte Sir Clive Sinclair, „unsere Probleme waren immer nur kurzfristiger Natur. So sehr wir Robert Maxwell dankbar waren für seine Unterstützung, freuen wir uns, als unabhängiges Unternehmen forzubestehen."

Das neue Management Team unter Chief Executive Bill Jeffrey sei jetzt etabliert, und es gebe eine Reihe interessanter neuer Produkte.

MONKEY DONG

TI-99/4A + Extended Basic

So rüde schlägt der Riesenaffe MONKEY DONG zu, wenn Sir Geoffrey versucht, seine Verlobte Miss Applebee aus der Gewalt des Monsters zu befreien!

MONKEY DONG – das Spiel – hat zwei Felder, ein Bonusfeld, neun(!) verschiedene Level und ein gigantisches Vorprogramm!

Joystick-Steuerung

Wer kennt sie nicht, die wohl phantastischste Geschichte, die je aus Hollywood zu uns drang: „King Kong und die weiße Frau". MONKEY DONG erzählt diese Story auf's Neue. Doch: ***Diese Spielversion ist einzigartig.*** Einfach anders.

Zunächst zur Story:
MONKEY DONG hat Miss Applebee entführt. Auf der Spitze eines Wolkenkratzer-Rohbaus thront er nun, der Riesenaffe, Miss Applebee umklammernd. Sir Geoffrey beginnt bei seinem Rettungsversuch von ganz unten, auf einem Eisenträger, Der wackere und verliebte Held muß nun über Leitern zu seiner Verlobten gelangen. Bei seinem Unterfangen behindern ihn die „Rolling Tuns" (rollende Tonnen!), ein Flammen-Meer sowie garstige Vögel. Für den adligen Held heißt dies: Ausweichen oder überspringen.

Auf seinem Weg kann Sir Geoffrey Schätze einsammeln, die die Bauarbeiter vergessen haben. (Er muß sich auf diese stellen und hüpfen. Im zweiten Feld besteht dann noch eine zusätzliche Schwierigkeit, weil der Held zwei Schlüssel ergattern muß, um Türen aufzusperren. Diese sind nur über einen Fahrstuhl zu erreichen. Schafft es nun Sir Geoffrey, seine Liebste zu Erretten, folgt nach einer rührenden Liebes-Szene eine Zeitgutschrift in Form von Bonuspunkten. Der Held darf nicht lange bei Miss Applebee verweilen, denn nun folgt die Aufgabe, vom Himmel fallende Geldsäcke zu sammeln. Alle 5000 Punkte bekommen Sie einen „Extra Adeligen"! Gesteuert mit dem Joystick 1; gesprungen wird mit dem Feuerknopf. (M.K.)

„Versetzen" Sie sich in Geoffrey's Lage: Retten Sie Ihre Herzensliebste in der Gestalt ● der „charming" Miss Applebee! ●

MONKEY DONG
Teil Eins – das VOR-PROGRAMM

... auf geht's!!

```
100 !
110 !        ***************
120 !        *MONKEY--DONG*
130 !        *VORPROGRAMM *
140 !        *  VER  1.2  *
150 !        ***************
160 !        *PROGRAMMIERT*
170 !        *     VON     *
180 !        *PETER HLAWNA*
190 !        * AM KOBEL 8 *
200 !        *82 ROSENHEIM*
210 !        ***************
220 !        *  (C) 1985   *
230 !        *     BY      *
240 !        *PETER'S SOFT*
250 !        ***************
260 ! CHARDEFINITIONEN
270 !
```

```
280 CALL CHAR(96,"0001030703030301070F1B1303060C080080C0F840E0C080E0F0DCC0C06030
18")
290 CALL CHAR(100,"0000010302070301030700D0103070F020000C0C0E0E0E080C0E0B080C0E0F
040")
300 CALL CHAR(104,"0070381E0F3F23000",106,"00000060D0FEE0C00")
310 CALL CHAR(124,"00000F3F23070E000",126,"60D0FEE0C08000000")
320 CALL CHAR(108,RPT$("F",16),112,RPT$("F",16),120,RPT$("F",16),128,"181818FFFF
181818")
330 CALL COLOR(10,16,1,11,7,1,12,4,1,13,10,2,4,8,8)
340 !
350 ! BILDSCHIRMAUFBAU
360 !
370 CALL CLEAR :: CALL SCREEN(2):: CALL MAGNIFY(3)
380 CALL HCHAR(1,1,57,640):: CALL HCHAR(20,1,120,160)
390 FOR I=11 TO 14 :: CALL HCHAR(I,9,112,13):: NEXT I
400 FOR I=15 TO 20 :: CALL HCHAR(I,10,108,11):: NEXT I
410 FOR I=16 TO 18 STEP 2 :: FOR J=11 TO 20 STEP 2 :: CALL HCHAR(I,J,128):: NEXT
 J :: NEXT I
420 CALL SPRITE(#1,96,15,160,100)
430 CALL SPRITE(#2,100,3,160,110)
440 !
450 ! MUSIK: "YESTERDAY"
460 !
470 RESTORE 480 :: FOR I=1 TO 31 :: READ A,B,C,D,E :: CALL SOUND(1.2*A,B,C,B+1,C
,D,E):: NEXT I
480 DATA 200,196,6,392,0
490 DATA 200,175,6,349,0
500 DATA 1200,175,6,349,0
510 DATA 400,123,6,147,6
520 DATA 200,123,7,440,0
530 DATA 200,123,9,494,0
540 DATA 200,139,6,554,0
550 DATA 200,139,7,587,0
560 DATA 200,139,9,659,0
570 DATA 200,139,13,698,0
580 DATA 300,110,6,659,0
590 DATA 100,110,7,587,0
600 DATA 1200,147,6,587,0
610 DATA 400,117,6,233,6
620 DATA 200,117,7,587,0
630 DATA 200,117,9,587,0
640 DATA 200,196,6,523,0
650 DATA 200,165,7,466,0
660 DATA 200,165,9,440,0
670 DATA 200,165,13,392,0
680 DATA 400,175,6,466,0
690 DATA 200,175,7,440,0
700 DATA 600,175,9,440,0
710 DATA 400,165,6,392,0
720 DATA 400,110,6,349,0
730 DATA 200,110,7,440,0
740 DATA 600,123,6,392,0
750 DATA 400,123,7,294,0
760 DATA 400,117,6,349,0
770 DATA 200,131,6,440,0
780 DATA 1000,131,10,440,0
790 !
800 ! ENTFUEHRUNG
810 !
820 CALL CHAR(136,"80888A8B8B8A8AFB8A8A8A8B8B8A8880000000000080D715971414D88")
830 CALL SPRITE(#3,104,12,115,1,3,10)
840 CALL POSITION(#3,Y,X):: IF X<105 THEN 840 :: Q=10
850 CALL LOCATE(#2,154,110):: CALL MOTION(#2,-5,10):: CALL MOTION(#3,-5,10):: CA
LL SPRITE(#4,136,5,152,98,-5,10)
860 Q=-Q :: CALL PATTERN(#3,114+Q)
870 FOR I=1 TO 50 :: NEXT I :: CALL POSITION(#3,Y,X):: IF X<240 THEN 860
880 CALL DELSPRITE(#2,#3,#4)
890 CALL CHAR(140,"0001030703030301030303030301010100080C0F840E0C080C0C0C0C0C0808
0C0")
900 Q=22 :: CALL MOTION(#1,0,9)
910 Q=-Q :: CALL PATTERN(#1,118+Q)
920 CALL POSITION(#1,Y,X)
930 FOR I=1 TO 40 :: NEXT I :: IF X>240 THEN CALL DELSPRITE(#1)ELSE 910
940 CALL COLOR(1,5,4,2,5,4,3,5,4):: FOR I=5 TO 8 :: CALL COLOR(I,5,4):: NEXT I
950 DISPLAY AT(21,4)SIZE(21):"M O N K E Y - D O N G" :: DISPLAY AT(23,7)SIZE(13)
:"WIRD  GELADEN"
960 !
970 ! JE NACH KONFIGURATION
980 ! ZEILE 1010 ODER 1020
990 ! EINGEBEN
1000 !
1010 RUN "CS1"
1020 RUN "DSK1.MONKEYII"
```

MONKEY-DONG – das VORPROGRAMM ist hier zu Ende. Teil 2 auf S. 42!

Fortsetzung von Seite 40!

MONKEY-DONG HAUPT-PROGRAMM

```
100 !        **************
110 !        *            *
120 !        *MONKEY--DONG*
130 !        *  VER  1.4  *
140 !        *            *
150 !        **************
160 !        *            *
170 !        *PROGRAMMIERT*
180 !        *     VON    *
190 !        *PETER HLAWNA*
200 !        * AM KOBEL 8 *
210 !        *82 ROSENHEIM*
220 !        *            *
230 !        **************
240 !        *            *
250 !        *  (C) 1985  *
260 !        *     BY     *
270 !        *PETER'S SOFT*
280 !        *            *
290 !        **************
300 !
310 CALL CHARSET :: CALL CLEAR :: K=2 :: Q,FELD=1 :: M=3 :: F=5000 :: PG,P=0 ::
CC=3000 :: DEF G(X)=INT(X/8)+1
320 !
330 ! ZEICHEN
340 !
350 ! ER,SIE
360 !
370 CALL CHAR(90,"00000103020703 01")
380 CALL CHAR(96,"0001031F02070301070F3B03030 60C180080C0E0C0C0C080C0F0D8C8C06030
10")
390 CALL CHAR(80,"0001031F02070301030303030301010300 80C0E0C0C0C080C0C0C0C0C08080
80")
400 CALL CHAR(104,"00010307030303010 70F1B1303060C080080C0F840E0C080E0F0DCC0C0603
018")
410 CALL CHAR(84,"00010307030303010303030303010101008 0C0F840E0C080C0C0C0C0C08080
C0")
420 CALL CHAR(100,"00010307030303111 71F0303030202060080C0E0C0C0C088E8F8C0C0C0404
060")
430 CALL CHAR(76,"00010307020303010 70F3B03233F00000080C0E040C0C080E0F0DCC0C4FC")
440 CALL CHAR(112,"03070D0103070F02C0E0B080C0E0F040")
450 CALL CHAR(127,"C0C0E0E0E080")
460 !
470 ! TONNEN,VOEGEL
480 !
490 CALL CHAR(120,"",121,"03070F0F0F0F07030",123,"C0E0F0F0F0F0E0C0")
500 CALL CHAR(64,"0000E0381E0F3F270",66,"00000060D0FEE0C00")
510 CALL CHAR(68,"000000060B7F07030",70,"0000071C78F0FCE40")
520 !
530 ! AFFE
540 !
550 CALL CHAR(128,"3F7EFCFCF8F8F0F0FC7E3F3F1F1F0F0FFFFFFFFFFFF7F3F3FFFFFFFFFFFEF
CFC")
560 CALL CHAR(134,"FFFFFFE7C3810000FFFFFFFFFFFFFFFF3CDBFFE7FF81817E")
570 CALL CHAR(132,"03071F3F3F3FFFFFC0F0F8FCFCFCFFFF")
580 !
590 ! LEITER,BAHN,HERZ
600 ! SCHAETZE,KREUZ
610 !
620 CALL CHAR(88,"FFFFC3C3FFFFC3C3")
630 CALL CHAR(115,"FFFFC33C3CC3FFFF")
640 CALL CHAR(108,"667EFF7E3C3C1818")
650 CALL CHAR(91,"3C3C7EFFFFFFFFFF")
660 CALL CHAR(92,"3C3C3CFFFFFFFF3C3C3C3C3C3C3C3C3C3C0",95,"")
670 CALL COLOR(10,10,1,9,16,1)
680 CALL COLOR(8,16,1,11,8,1)
690 CALL COLOR(12,10,1,13,7,1,14,15,1)
700 CALL CLEAR :: CALL SCREEN(2):: CALL MAGNIFY(3):: RANDOMIZE
710 !
720 FOR I=1 TO 7 :: CALL COLOR(I,16,1):: NEXT I
730 !
740 ! BILDSCHIRMAUFBAU
```

```
750 !
760 CALL DELSPRITE(ALL):: GOSUB 1620 :: GOSUB 1590
770 ! BAHN
780 FOR I=8 TO 24 STEP 4
790 CALL HCHAR(I,1,115,32):: NEXT I :: CALL HCHAR(4,10,115,14)
800 ! LEITER
810 RESTORE 820 :: FOR I=1 TO 9 :: READ A,B :: CALL VCHAR(A,B,88,4):: NEXT I
820 DATA 2,24,4,24,8,5,8,19,12,26,16,8,16,17,20,25,4,13
830 ! SCHAETZE
840 FOR I=1 TO 6
850 READ A,B
860 CALL HCHAR(A,B-1,32,3):: CALL HCHAR(A,B,91):: NEXT I
870 DATA 7,18,11,14,15,6,15,28,19,6,23,28
880 ! AFFE
890 FOR I=5 TO 7 :: FOR J=28 TO 30 :: READ A :: CALL HCHAR(I,J,A):: NEXT J :: NE
XT I
900 DATA 128,136,129,130,135,131,132,134,133
910 ! SIE
920 CALL HCHAR(2,17,90):: CALL HCHAR(2,18,127):: CALL HCHAR(3,17,112):: CALL HCH
AR(3,18,113)
930 ! TONNEN
940 D=MIN(3,K)
950 C=CC :: T=1 :: A=50 :: FOR I=40 TO 168 STEP 32 :: A=-A :: FOR J=1 TO D :: T=
T+1 :: CALL SPRITE(#T,120,10,I,256,0,A)
960 FOR L=1 TO (250/D):: NEXT L :: NEXT J :: NEXT I
970 T=1 :: A=5 :: FOR I=1 TO 5 :: A=-A :: FOR J=1 TO D :: T=T+1 :: CALL MOTION(#
T,0,A):: NEXT J :: NEXT I
980 IF K>3 THEN GOSUB 3030
990 M=M-1 :: IF M<0 THEN 1370
1000 !
1010 ! SPIELBEGINN
1020 !
1030 CALL JOYST(1,XJ,YJ):: IF XJ+YJ=0 THEN 1030 ELSE GOSUB 1590
1040 CALL SPRITE(#1,104,15,168,16)
1050 CALL KEY(1,KEY,ST):: CALL JOYST(1,XJ,YJ):: CALL MOTION(#1,0,2*XJ)
1060 IF KEY=18 THEN GOSUB 1270
1070 IF YJ<>0 THEN GOSUB 1140
1080 Q=-Q :: IF XJ=-4 THEN CALL PATTERN(#1,88+8*Q)ELSE IF XJ=4 THEN CALL PATTERN
(#1,94+10*Q)
1090 CALL COINC(ALL,Z):: CALL POSITION(#1,Y,X):: IF X<8 OR X>250 OR Z THEN 1550
1100 C=C-5 :: GOTO 1050
1110 !
1120 ! KLETTERN
1130 !
1140 CALL PATTERN(#1,100):: CALL MOTION(#1,0,0)
1150 CALL POSITION(#1,Y,X):: IF YJ=-4 THEN 1160 ELSE CALL GCHAR(G(Y),G(X+7),A):
 IF A<>88 THEN RETURN ELSE 1170
1160 IF Y>152 THEN 1090 ELSE CALL GCHAR(G(Y)+2,G(X+7),A):: IF A<>88 THEN RETURN
1170 CALL MOTION(#1,-YJ,0):: FOR I=1 TO 20 :: CALL COINC(ALL,A):: IF A THEN ON F
ELD GOTO 1550,2740
1180 NEXT I
1190 CALL POSITION(#1,Y,X):: C=C-5 :: CALL COINC(ALL,A):: IF A=-1 THEN 1550
1200 CALL GCHAR(G(Y)+2,G(X-1),A)
1210 IF FELD=2 THEN GOSUB 2790
1220 IF A=115 THEN 1230 ELSE CALL JOYST(1,XJ,YJ):: CALL MOTION(#1,-YJ,0):: GOTO
1190
1230 CALL MOTION(#1,0,0):: CALL LOCATE(#1,G(Y)*8-8,X):: IF Y<20 THEN 1680 ELSE R
ETURN
1240 !
1250 ! SPRUNG
1260 !
1270 CALL PATTERN(#1,76):: CALL POSITION(#1,Y,X):: CALL LOCATE(#1,Y-8,X):: CALL
GCHAR(G(Y)+1,G(X+7),A):: IF A=91 THEN GOSUB 1300
1280 FOR I=1 TO 30 :: CALL COINC(ALL,A):: IF A THEN ON FELD GOTO 1550,2740
1290 NEXT I :: C=C-75 :: CALL POSITION(#1,Y,X):: CALL PATTERN(#1,84):: CALL LOCA
TE(#1,Y+8,X):: RETURN
1300 P=INT(RND*5)*100+100
1310 GOSUB 1620
1320 CALL HCHAR(G(Y)+1,G(X),P/100+48):: CALL HCHAR(G(Y)+1,G(X)+1,48,2)
1330 FOR J=1 TO 3 :: CALL SOUND(100,440,0):: CALL SOUND(100,550,0):: CALL SOUND(
100,660,0):: NEXT J :: RETURN
```

```
1340 !
1350 ! SCHLUSS?
1360 !
1370 CALL DELSPRITE(ALL)
1380 CALL CLEAR :: CALL CHARSET
1390 FOR J=1 TO 8 :: CALL COLOR(J,16,1):: NEXT J
1400 DISPLAY AT(10,11):"GAME OVER"
1410 DISPLAY AT(14,8):"SCORE     :";PG :: DISPLAY AT(15,8):"HIGHSCORE:";HI
1420 DISPLAY AT(18,5):"NOCH EINMAL? (J/N) J"
1430 ACCEPT AT(18,24)VALIDATE("JNJn")SIZE(-1):A$
1440 IF A$="J" OR A$="j" THEN 310
1450 END
1460 !
1470 ! FREIFIGUR
1480 !
1490 F=F+5000 :: M=M+1
1500 DISPLAY AT(1,1)SIZE(8):RPT$("hJ",M):: DISPLAY AT(2,1)SIZE(8):RPT$("ik",M)
1510 RETURN
1520 !
1530 ! MAENNCHEN KAPUTT
1540 !
1550 CALL MOTION(#1,0,0):: CALL PATTERN(#1,92):: CALL COLOR(#1,16)
1560 CALL SOUND(300,440,0):: CALL SOUND(300,330,0):: CALL SOUND(300,220,0):: CAL
L SOUND(500,110,0)
1570 CALL MOTION(#1,-7,0):: CALL POSITION(#1,Y,X):: IF Y>5 THEN 1570 ELSE CALL D
ELSPRITE(#1):: IF PG>F THEN GOSUB 1470
1580 ON FELD GOTO 990,2770
1590 ! ANZEIGE DER MAENNCHEN
1600 DISPLAY AT(1,1)SIZE(8):RPT$("hJ",M):: DISPLAY AT(2,1)SIZE(8):RPT$("ik",M)::
 RETURN
1610 ! PUNKTANZEIGE
1620 PG=PG+P :: DISPLAY AT(1,9):PG:: HI=MAX(PG,HI)
1630 DISPLAY AT(1,20):"HI:";HI :: IF PG>=F THEN GOSUB 1470
1640 RETURN
1650 !
1660 ! 1.FELD GESCHAFFT
1670 !
1680 IF FELD=2 THEN 2840 ELSE FELD=2
1690 IF X>120 THEN 1720
1700 CALL MOTION(#1,0,8):: CALL PATTERN(#1,84)
1710 CALL POSITION(#1,Y,X):: IF X<120 THEN 1710 ELSE 1740
1720 CALL MOTION(#1,0,-8):: CALL PATTERN(#1,80)
1730 CALL POSITION(#1,Y,X):: IF X>120 THEN 1720 ELSE CALL PATTERN(#1,84)
1740 CALL MOTION(#1,0,0)
1750 FOR J=1 TO 3 :: CALL SOUND(100,440,0):: CALL SOUND(100,550,0):: CALL SOUND(
100,660,0):: CALL SOUND(100,770,0):: NEXT J
1760 P=MAX(C,0):: GOSUB 1620 :: DISPLAY AT(4,23):P :: CC,C=CC+2000
1770 FOR J=1 TO 3 :: CALL HCHAR(1,17,108):: FOR L=1 TO 200 :: NEXT L :: CALL HCH
AR(1,17,32):: FOR L=1 TO 100 :: NEXT L :: NEXT J
1780 GOSUB 2920
1790 !
1800 ! 2. FELD
1810 !
1820 ! ZEICHEN
1830 !
1840 ! LIFT,SCHLUEESEL
1850 !
1860 CALL CHAR(60,"",61,"80808080808080FF0",63,"02020202020202FE")
1870 CALL CHAR(40,"0303030303030303")
1880 CALL CHAR(89,"1C22221C08381838")
1890 !
1900 ! FLAMMEN,TUER
1910 !
1920 CALL CHAR(64,"",65,"FFFFFFFFFF7E7E3C0",67,"")
1930 CALL CHAR(33,"0003070F1F3F7F7F00C0E0F0F8FCFEFEFFFFFFFFFFFFFFFF")
1940 FOR I=2 TO 7 :: CALL COLOR(I,16,1):: NEXT I
1950 CALL COLOR(1,7,1,2,12,1)
1960 XX=-3 :: M=M+1 :: P=0 :: Q1=1
1970 !
1980 ! BILDSCHIRMAUFBAU
1990 !
```

```
2000 CALL CLEAR :: CALL DELSPRITE(ALL):: GOSUB 1620 :: GOSUB 1590
2010 ! BAHN
2020 RESTORE 2030 :: FOR I=1 TO 8 :: READ A,B,C :: CALL HCHAR(A,B,115,C):: NEXT
I
2030 DATA 4,20,11,8,18,15,12,9,24
2040 DATA 14,1,4,16,7,26,20,7,26,24,7,26,5,1,4
2050 ! LEITERN
2060 FOR I=1 TO 5 :: READ A,B :: CALL VCHAR(A,B,88,4):: NEXT I
2070 DATA 4,21,8,29,12,20,20,11,20,22
2080 ! TUEREN
2090 CALL HCHAR(9,28,33):: CALL HCHAR(9,29,35):: CALL HCHAR(9,30,34)
2100 CALL HCHAR(10,28,35,3):: CALL HCHAR(11,28,35,3)
2110 CALL HCHAR(5,20,33):: CALL HCHAR(5,21,35):: CALL HCHAR(5,22,34)
2120 CALL HCHAR(6,20,35,3):: CALL HCHAR(7,20,35,3)
2130 ! SCHLUESSEL
2140 CALL HCHAR(4,2,89):: CALL HCHAR(13,2,89)
2150 ! AUFZUG
2160 CALL VCHAR(3,5,40,22)
2170 CALL SPRITE(#2,60,12,184,33,-4,0)
2180 ! SIE
2190 CALL HCHAR(2,24,90):: CALL HCHAR(2,25,127):: CALL HCHAR(3,24,112):: CALL HC
HAR(3,25,113)
2200 ! AFFE
2210 FOR I=5 TO 7 :: FOR J=30 TO 32 :: READ A :: CALL HCHAR(I,J,A):: NEXT J :: N
EXT I
2220 DATA 128,136,129,130,135,131,132,134,133
2230 ! FLAMMEN
2240 FOR I=3 TO 5
2250 READ A,B :: CALL SPRITE(#I,64,10,A,B,0,XX):: NEXT I
2260 DATA 168,152,104,120,72,220
2270 IF K=2 THEN 2330
2280 RESTORE 2290 :: FOR I=6 TO 8 :: READ A,B :: CALL SPRITE(#I,64,10,A,B,0,XX):
: NEXT I
2290 DATA 136,180,72,150,40,190
2300 IF K=3 THEN 2330
2310 RESTORE 2320 :: FOR I=9 TO 11 :: READ A,B :: CALL SPRITE(#I,64,10,A,B,0,XX)
:: NEXT I
2320 DATA 168,210,104,220,136,80
2330 GOSUB 2790
2340 IF M=0 THEN 1370
2350 !
2360 ! SPIELBEGINN
2370 !
2380 CALL JOYST(1,XJ,YJ):: IF XJ+YJ=0 THEN 2330
2390 M=M-1 :: GOSUB 1590
2400 CALL SPRITE(#1,80,15,168,230)
2410 CALL KEY(1,KEY,ST):: CALL JOYST(1,XJ,YJ):: CALL MOTION(#1,0,2*XJ)
2420 IF YJ THEN GOSUB 1140
2430 IF KEY=18 THEN GOSUB 1270 :: GOTO 2450
2440 Q=-Q :: IF XJ=-4 THEN CALL PATTERN(#1,88+8*Q)ELSE IF XJ=4 THEN CALL PATTERN
(#1,94+10*Q)
2450 CALL COINC(ALL,A):: CALL POSITION(#1,Y,X):: IF A OR X<38 OR X>250 THEN 2740
2460 GOSUB 2790 :: C=C-5 :: IF RND<.1 THEN GOSUB 3110
2470 GOTO 2410
2480 !
2490 ! IM AUFZUG
2500 !
2510 CALL PATTERN(#1,80):: CALL MOTION(#1,-4,0):: CALL POSITION(#2,Y2,X2):: CALL
 LOCATE(#1,MAX(Y2-2,1),32):: FOR I=1 TO 200 :: NEXT I
2520 CALL JOYST(1,XJ,YJ)
2530 IF XJ<>0 THEN 2560
2540 GOSUB 2790
2550 GOTO 2520
2560 CALL MOTION(#1,0,0):: CALL POSITION(#1,Y,X):: IF XJ=-4 THEN 2590 ELSE CALL
PATTERN(#1,84):: CALL LOCATE(#1,Y,X+20)
2570 IF Y>170 OR Y<90 THEN 2740 ELSE IF Y<155 AND Y>138 OR Y<120 AND Y>112 THEN
2740
2580 CALL LOCATE(#1,INT(Y/32)*32+8,X+20):: GOTO 2410
2590 CALL PATTERN(#1,80):: CALL LOCATE(#1,Y,X-16)
2600 IF Y>100 OR Y<85 THEN 2650 ELSE CALL LOCATE(#1,11*8,X-16)
2610 CALL GCHAR(13,2,A):: IF A=89 THEN P=INT(RND*5)*100+100 :: C=C+1000 ELSE 269
```

ENDSPURT: RETTEN SIE APPLEBEE!

```
0
2620 CALL HCHAR(13,2,P/100+48):: CALL HCHAR(13,3,48,2)
2630 GOSUB 1620 :: GOSUB 1330 :: IF PG>=F THEN GOSUB 1470
2640 CALL VCHAR(9,28,32,3):: CALL VCHAR(9,29,88,3):: CALL VCHAR(9,30,32,3):: GOT
O 2690
2650 IF Y>37 OR Y<12 THEN 2740 ELSE CALL LOCATE(#1,16,X-16)
2660 CALL GCHAR(4,2,A):: IF A=89 THEN P=INT(RND*5)*100+100 :: C=C+1000 :: CALL H
CHAR(4,2,P/100+48):: CALL HCHAR(4,3,48,2)
2670 GOSUB 1620 :: GOSUB 1330 :: IF PG>=F THEN GOSUB 1470
2680 CALL VCHAR(5,20,32,3):: CALL VCHAR(5,21,88,3):: CALL VCHAR(5,22,32,3)
2690 CALL KEY(1,KEY,ST):: CALL JOYST(1,XJ,YJ):: CALL MOTION(#1,0,2*XJ)
2700 IF KEY=18 THEN GOSUB 1270
2710 GOSUB 2790 :: IF RND<.1 THEN GOSUB 3110
2720 CALL POSITION(#1,Y,X):: IF X<4 OR X>23 THEN 2740 ELSE 2690
2730 ! MAENNCHEN KAPUTT 2
2740 CALL COINC(#1,#2,10,A):: IF A=-1 THEN 2510
2750 CALL MOTION(#1,0,0):: FOR J=3 TO 11 :: CALL MOTION(#J,0,0):: NEXT J
2760 GOTO 1550
2770 FOR J=3 TO 11 :: CALL MOTION(#J,0,XX):: NEXT J :: GOTO 2380
2780 ! FLAMMEN UMDREHEN?
2790 CALL POSITION(#3,Y3,X3):: IF X3<104 THEN XX=4 ELSE IF X3>152 THEN XX=-4 ELS
E RETURN
2800 FOR J=3 TO 11 :: CALL MOTION(#J,0,XX):: NEXT J :: RETURN
2810 !
2820 ! 2.FELD GESCHAFFT
2830 !
2840 FELD=1 :: FOR I=3 TO 11 :: CALL MOTION(#I,0,0):: NEXT I
2850 CALL MOTION(#1,0,8):: CALL PATTERN(#1,84)
2860 CALL POSITION(#1,Y,X):: IF X<174 THEN 2860
2870 CALL MOTION(#1,0,0):: FOR I=1 TO 3 :: CALL SOUND(100,440,0):: CALL SOUND(10
0,550,0):: CALL SOUND(100,660,0):: CALL SOUND(100,770,0)
2880 NEXT I :: P=C :: GOSUB 1620 :: DISPLAY AT(4,24):C
2890 FOR J=1 TO 3 :: CALL HCHAR(1,24,108):: FOR L=1 TO 200 :: NEXT L :: CALL HCH
AR(1,24,58):: FOR L=1 TO 100 :: NEXT L :: NEXT J
2900 K=K+1 :: M=M+1 :: GOSUB 2920 :: GOTO 370
2910 !
2920 ! ZWISCHENSPIEL
2930 !
2940 CALL CLEAR :: CALL DELSPRITE(ALL):: CALL HCHAR(22,1,115,32):: P=0 :: GOSUB
1620 :: X1=5 :: P=100 :: DISPLAY AT(8,13):P
2950 GOSUB 1590 :: FOR I=1 TO 8 :: CALL COLOR(I,16,1):: NEXT I :: CALL SPRITE(#1
,100,15,152,126)
2960 CALL CHAR(120,"000303010307070F1F1F1F1F1F0F070300C0C080C0E0E0F0F8F8F8F8F8F0
F0E0C0")
2970 CALL SPRITE(#2,120,INT(RND*14)+3,1,INT(RND*200)+25,X1,0)
2980 CALL JOYST(1,XJ,YJ):: CALL MOTION(#1,0,5*XJ)
2990 CALL COINC(#1,#2,10,A):: IF A=-1 THEN 3010 ELSE CALL POSITION(#2,Y,X):: IF
Y<171 THEN 2980
3000 P=0 :: CALL DELSPRITE(ALL):: RETURN
3010 CALL DELSPRITE(#2):: GOSUB 1620 :: P=P+10 :: X1=X1+2 :: DISPLAY AT(8,13)BEE
P:P :: GOTO 2970
3020 ! VOEGEL
3030 T=17 :: FOR I=30 TO 126 STEP 64 :: FOR J=1 TO K-3 :: CALL SPRITE(#T,64,12,I
,1,0,50):: T=T+1
3040 FOR L=1 TO (280/(K-3)):: NEXT L :: NEXT J :: NEXT I
3050 FOR I=17 TO 22 :: CALL MOTION(#I,0,10):: NEXT I
3060 T=23 :: FOR I=62 TO 158 STEP 64 :: FOR J=1 TO K-3 :: CALL SPRITE(#T,68,12,I
,1,0,-50):: T=T+1
3070 FOR L=1 TO (280/(K-3)):: NEXT L :: NEXT J :: NEXT I
3080 FOR I=23 TO 28 :: CALL MOTION(#I,0,-10):: NEXT I
3090 RETURN
3100 ! FLAMMEN EIN/AUS
3110 Q1=-Q1 :: IF Q1=1 THEN CALL CHAR(64,"")ELSE CALL CHAR(64,"0000081818 1C4D77")
3120 RETURN
```

Vier „goldene" Tips für Ihren TI!

Er ist zwar schon etwas „in die Jahre gekommen", der TI-99/4A. Dennoch gibt es viele User „in diesem, unseren Lande" (aber auch im Ausland), die den TI zu schätzen wissen. Deshalb hat die HOMECOMPUTER-Redaktion hier einige Tips zusammengestellt. (M.K.)

1) Der TI Sprachsynthesizer läßt sich von Extended Basic aus bekanntlich über CALL SAY ansprechen. Dazu nun einige Tips:
Das Ungleich- bzw. Number-Zeichen # ermöglicht ein schnelleres und zusammenhängenderes Sprechen. Am Anfang sollte es nur einmal stehen (der Computer „verschluckt" sonst Buchstaben), zwischen den Worten je zweimal.
Beispiel: 100 CALL SAY("#PRESS## ANY##KEY")
Mehr als zwei # - Zeichen ermöglichen ein beliebiges Wiederholen von Worten.
Beispiel: 100 CALL SAY("#A###")
Außerdem klingen eine Reihe von Worten bzw. Buchstaben gleich, so daß man hier auch Speicherplatz sparen kann.

-	=	negative
+	=	positive
.	=	point
M	=	am
S	=	as

B	=	be
BY	=	buy
I	=	eye
W	=	you
y	=	why ...

Außerdem können statt ohne, two, etc. die Zahlen auch direkt als 1, 2, ... eingegeben werden.
Beispiel: Statt CALL SAY("I AM, YOU ARE") nur CALL SAY(" I M, U R")

★ ★ ★

2) Vielen TI-Basic Programmierern ist gar nicht bewußt, was für Extended Basic-Fans selbstverständlich ist: Um eine Zeilennummer zu editieren, muß man auch in TI-Basic nicht EDIT Zeilennummer eingeben! Es reicht, die Taste FCTN und gleichzeitig X oder E zu drücken, und schon ist man im Edit-Modus. Im Extended Basic gibt es den Befehl EDIT übrigens gar nicht.

★ ★ ★

3) Eine Reihe von TI-Usern besitzen keine Joysticks und schreiben deshalb die Programme als Zeitschriften um, damit Sie die Tastatur verwenden können. Dabei wird oft unnötige Arbeit geleistet. Denn die Abfrage der Fire-Taste des Joystick gilt gleichzeitig für die Tastatur. Fire-Taste Eins liefert den gleichen Code wie Q, Fire-Taste Zwei die gleichen Code wie Y. Auch bei Modulen können Q bzw. Y anstatt der Fire-Tasten verwendet werden.

★ ★ ★

4) Und nun noch einige Tips für echte Insider, „Hacker" und Programmierfreaks: Wenn Ihr ein echtes Superspiel entwikkelt, dann wird oft fast der gesamte Speicherbereich des TI-99 belegt, und irgendwann (kurz vor Schluß ...) bricht das Programm ab: MEMORY FULL! Jetzt hilft nur eins: Kürzen, wo es nur geht. Daß zunächst alle REM-Zeilen gelöscht werden müssen, ist klar. Auch die Print-Zeilen mit der Spielanleitung o.ä. müssen gekürzt werden, denn sie brauchen viel Speicherplatz. Wenn das Programm dann immer noch nicht läuft, wird es problematisch. Sollten SUB-Unterprogramme enthalten sein, so sollte man versuchen, diese durch Gosub-Return-Unterprogramme zu ersetzen.

Denn: Alle Variablen in einem SUB - Unterprogramm (eine sogenannte „Prozedur") sind lokale Variablen. Sie werden wie völlig neue Variablen behandelt; sogar wenn Variablen mit dem gleichen Namen im Hauptprogramm vorkommen. Also Vorsicht beim Kürzen! Ein weiterer Schritt ist es, die Namen der Variablen zu kürzen. Aus MONAT=12 wird M=12, aus WOCHE=52 wird W=52, etc.. Auch hier ist Vorsicht geboten, damit man einen Buchstaben nicht zweimal verwendet.

Und hier nun der eigentliche Trick:
Wenn man, wie eben geschildert, den Namen einer Variablen ändern will, so muß man wirklich alle Zeilen, in denen diese Variable vorkommt, kennen. Und wer einmal ein meterlanges Programm untersucht hat, dem wird eine Variable wie die oft zitierte Nadel im Heuhaufen vorkommen. Leider fehlt dem TI-99 natürlich auch ein Befehl, der alle Zeilennummern ausgibt, in der eine bestimmte Variable vorkommt. Deshalb gehen Sie am besten wie folgt vor:
Nehmen wir einmal an, Sie suchen alle Zeilen mit der Variablen PUNKTESTAND, um sie beispielsweise durch P zu ersetzen. Dann geben Sie ein:
1 PUNKTESTAND(1)=1
Bacis-Kenner werden den Trick vielleicht schon verstanden haben. Der Computer läßt es nämlich nicht zu, eine Variable in verschiedenen Dimensionen mit dem gleichen Namen zu versehen. Wenn nun RUN eingegeben wird, so bricht der Computer sofort ab und gibt die erste Zeilennummer an, in der die Variable PUNKTESTAND vorkommt. Denn der TI hat PUNKTESTAND ja schon als Feld verstanden, nicht als einfache Variable. Nun ändern Sie die Variable in der entsprechenden Zeile in P um und geben wieder RUN ein. Der Computer nennt Ihnen die nächste Zeile und so weiter. Sobald das Programm nicht mehr abbricht, haben Sie alle Variablen „umgetauft". Nun müssen Sie nur noch Zeile 1 löschen, fertig!

Übrigens, falls Sie eine dimensionierte Variable umnennen wollen, geben Sie in Zeile 1 den gleichen Namen einfach wie eine normale Variable ein.
Beispiel: ERGEBNIS(10) in E(10)
Zeile 1: ERGEBNIS=1 : Das Programm bricht bei allen ERGEBNIS(X) ab!

HOMECOMPUTER sucht noch weitere TIPS & TRICKS! Schreibt uns doch!

„Hereinspaziert, meine Damen und Herren, liebe Kinder! Attraktion folgt auf Attraktion. Hier ist er wieder, der

CIRCUS TI-99/4A – Extended Basic

Erleben Sie den Zauber der Manege! Akrobatik, Spaß und Unterhaltung: Für Stimmung und Spielwitz ist gesorgt!

„Manege frei!" heißt es bei diesem abwechslungsreichen und hochinteressanten Spiel. Ihre Geschicklichkeit im Umgang mit Trampolinturnern, Bogenschützen und bei Tier-Dressuren ist gefragt. CIRCUS besteht aus vier Übungen, die erst einmal erfolgreich bewältigt werden wollen. *(M.K.)*

„Circus hin, Circus her! Im Straßenverkehr können Sie sich eben solche Scherze nicht erlauben!"

1. Die 3 Trampolinos.
Eine gewagte 3-Personen Übung, bei der mit dem Joystick Nr. 1 abwechselnd der Springer bzw. die beiden Fänger und das Trampolin gesteuert werden.
Nachdem der Springer vom Sprungturm abgesprungen ist, übernehmen Sie mit dem Joystick Nr. 1 die beiden Fänger mit ihrem Trampolin. Haben Sie den Springer gut aufgefangen, wechselt der Joystick auf den Springer über. Eine Steuerung nach links und rechts ist jetzt möglich. Hat der Springer mit seinen Händen einen Luftballon erreicht, müssen Sie durch Druck auf die Feuertaste den Ballon zerstechen. Es beginnt danach sofort wieder der Abwärtsfall, (der nicht immer gerade ist!). Sie steuern jetzt wieder die beiden Fänger.
Der Auftritt der „3 Trampolinos" ist zuende, wenn alle Ballons zerstochen wurden. Hat der Springer das Trampolin verfehlt, zappelt er noch eine Weile auf dem Boden herum, um danach symbolisch von einem Besen weggekehrt zu werden. Nach jeder gelungenen Darbietung haben Sie eine Pause, die von den Circusmusikern mit einer flotten Melodie untermalt wird. Sie können die Pause durch Druck auf den Feuerknopf beenden.

2. Bogen-Harry, der Scharfschütze.
In der Manege stehen 6 Kisten, aus denen Luftballons freigelassen werden. „Bogen-Harry" darf die Leiter des Sprungturms zum Rauf- und Runterklettern benutzen. Oben ist natürlich Schluß. Durch geschicktes Zielen muß er alle Ballons mit einem Pfeil zerschießen. (Alpha-Lock-Taste muß oben sein)

3. „Flips und Flaps" die beiden Seehunde.
Hier hat der Joystick wieder eine Doppelfunktion. Es wird immer der Seehund gesteuert, der den Ball fangen muß. Zu dieser Übung gehört ein gutes Augenmaß, denn der Seehund darf den Ball nur mit seinem Maul auffangen. Also rechtzeitig zu dem Aufschlagpunkt des Balles hinwatscheln. Natürlich muß der fangende Seehund im Moment der Ballannahme seinen Partner sehen. (Nicht die kalte Schulter zeigen!)
Fällt der Ball zu Boden, springt er noch eine Weile weg, um dann zu zerplatzen. Der Besen kehrt ihn dann wieder symbolisch zusammen.

4. Freddy und sein Pudel „Blacky".
Freddy hält eine große Tonne über seinem Kopf, durch die sein Hund „Blacky" springen muß, ohne die Tonne oder Freddy dabei zu berühren.
Da „Blacky" mit unterschiedlichen Geschwindigkeiten angelaufen kommt, ist es für Freddy (Sie als der Spieler sind jetzt gefordert) immer sehr schwierig, dem Hund den richtigen Absprung (Druck auf Feuertaste) zu geben. Gelingt ein Durchsprung, darf Freddy einen Schritt in Richtung Leiter machen. Hat er die Leiter erreicht, ist auch diese Übung absolviert. Damit ist ein Level beendet. Das Spiel beginnt nun mit dem nächsten höheren Level, bei dem alles ein wenig schneller geht.

Wichtig für die Eingabe:

1. Die in Programmzeile 430 ausgedruckten Buchstaben „D" und „F" sind bei gleichzeitig gedrückter DTRL-Taste einzugeben. Da es sich um frei definierte Zeichen handelt, werden sie erst nach „RUN" des Spieles sichtbar.
2. Die Zeilen 10–37 nicht eingeben, da sonst der Speicher ohne Erweiterung nicht reicht.
3. In Zeile 520 das „ON ERROR 1350" erst einfügen, wenn das Programm ohne erkennbare Fehler läuft.
4. Die ungeraden REM-Zeilen nicht miteingeben, da sonst der Speicher ohne Erweiterung nicht reicht. Die REM-Zeilen dienen zum besseren Erkennen der einzelnen Programm-Module.

```
10 !************************
11 !*     =============    *
12 !*    ==C I R K U S==   *
13 !*     =============    *
14 !*                      *
15 !*      COPYRIGHT BY    *
16 !*      W.H.DOELTSCH    *
17 !*                      *
18 !*  6108 WEITERSTADT 3  *
19 !*  ------------------  *
20 !*  A.D.HINTERSTEIN 10  *
21 !*   TEL.06150-40547    *
22 !************************
23 !*  BENOETIGTE GERAETE: *
24 !*  TI 99/4A - KONSOLE  *
25 !*   EXTENDED - BASIC   *
26 !*     U.RECORDER       *
27 !************************
28 !*   SPEICHERBELEGUNG   *
29 !*  NACH RUN UND BREAK  *
30 !*  CA.320 BYTES FREE!  *
31 !************************
32 !*    A C H T U N G!!   *
33 !* DIE ZEILEN 10-37 SIND*
34 !* NUR INFORMATORISCH  *
35 !* BITTE NICHT MIT EIN- *
36 !* GEBEN.              *
37 !************************
100 LEV=1 :: DIM S$(6),S(7),V(2):: CALL
CLEAR :: CALL SCREEN(2):: RANDOMIZE :: G
OTO 400
110 CALL SPRITE :: CALL COLOR :: CALL DI
STANCE :: CALL COINC :: CALL MOTION :: C
ALL LOCATE :: CALL DELSPRITE :: CALL HCH
AR
120 CALL CHAR :: CALL POSITION :: CALL K
EY :: CALL PATTERN :: CALL SOUND
130 A,B,C,D,E,F,G,R,ZE,HIT,HI,T,ST,Z,SP,
X,Y,LI,PU,HIG,I,LEV
140 !@P-
150 DATA 1,1,76,1,2,74,1,31,75,1,32,76,2
,1,76,2,2,74,2,31,75,2,32,76,3,1,76,3,2,
74,3,31,75
160 DATA 3,32,76,4,1,76,4,2,74,4,31,75,4
,32,76,5,1,76,5,2,74,5,31,75,5,32,76,6,1
,76,6,2,74
170 DATA 6,31,75,6,32,76,7,1,76,7,2,74,7
,31,75,7,32,76,8,1,76,8,2,74,8,31,75,8,3
2,76,9,1,76
180 DATA 9,2,74,9,31,75,9,32,76,10,1,76,
10,2,74,10,31,75,10,32,76,11,1,76,11,2,7
4,11,31,75,11,32,76
190 DATA 7,16,100,7,17,111,7,18,101,7,19
,108,7,20,116,7,21,115,7,22,99
200 DATA 12,1,76,12,2,74,12,31,75,12,32,
76,13,1,72,13,2,73,13,31,72,13,32,73,15,
2,93,15,31,92
210 DATA 16,1,93,16,2,94,16,31,95,16,32,
92,17,1,94,17,32,95,18,1,88,18,2,88,18,3
1,88,18,32,88
220 DATA 7,11,98,7,12,121,7,14,119,21,1,
88,21,2,88,21,31,88,21,32,88,7,23,104
230 DATA 256,200,40,43,1,152,96,70,72,2,
200,184,40,43,3
235 !*
240 DATA 2,440,2,587,740,587,659,740,784
,659,2,587,740,587,659,740
250 DATA 784,659,587,740,880,1175,1109,9
88,880,784,740,784,880,988,880,784
260 DATA 740,659,2,587,740,587,659,740,7
84,659,2,587,740,587,659,740
270 DATA 784,659,587,740,880,1175,1109,9
88,880,784,740,784,659,740,2,587,99
275 !*
280 DATA 370,392,440,370,294,370,440,370
,294,370,2,440,2,587,2,587
290 DATA 554,494,440,370,294,370,440,370
,294,370,2,392,2,330,2,330,370,392
300 DATA 440,370,294,370,440,370,294,370
,2,440,2,587,2,587,659,784,740
```

Spiel, Spaß, Spannung! im

CIRCUS

```
310 DATA 880,740,587,659,784,659,554,2,5
87,2,587,2,587,99
315 !*
320 DATA 36,81FF81FF81FF81FFFFFFFFFF3F0F
0301FFFFFFFF00000000081C3A59999A5C3FF,40,
181B0B3E9A1924621B18081C3A4B1434181B0B38
181B04141B1B103C59982446
330 DATA 44,181B10385C1A282C1B1B101C1B18
202800000000FF423C423C66DF9F9FCF7E3C,64,
02070522725727703048783078FCFCC00C1E2D3F
120C1E3F98527DBC7C6C9324
340 DATA 68,BAAA927C102844C63828107C92AA
286C3838127C9028680C3838907C12282C6,72,F
FFFFF7F3F1F0F03FFFFFFFEFCFBF0C0FFFBFDFEF
FFBFDFEFFDFBF7FFFDFBF7F
350 DATA 76,FFFFFFFFFFFFFFFFFFFFFFFFFFFF
FCE0FBFBF0E0C0000000FFFFFFFEFEFEFCFC,80,
FFFFFF7F7F7F3F3F1F1F0F0703000000FFFFFFFF
FFFF3F0710107C10101038FE
360 DATA 84,002060BC7F3E22142062A17D3E3E
226670A9613E3E3E42812062A17D3E3E2266
370 DATA 88,FFFFFFFFFFFFFFFF2041BE412000
00000000004121120DCFC0000001212D2DC3C,92,
B0C0E0F0FBFCFEFF0103070F1F3F7FFFFFFEFCFB
F0E0C0B0FF7F3F1F0F070301
380 DATA 128,01020468902B081010101010107
C5454B0402016091410081010101010107C5454
390 DATA 132,B0C0C0607C3E1E33406060303C3
E3F61010303063E7C78CC0206060C3C7CFCB6,13
6,1F3F7F7F7FFFFFFFFFFFFF7F7F7F3F1FFBE4C2
C2C2818181818181C2C2C2E4F8
395 !*
400 S$(1)=RPT$("KJ",30)&"KOHIHIHIHIHIHIH
IHIPJKJKJKJKJMN  JXXXXXXXXXXX\ QRKJKJKJK
JO   Xc i r c u sX  PKJKJKJKMN   _XXXXXX
XXXXX^  QRJKJ"
410 S$(2)="KJKO"&RPT$(" ",20)&"PJKJKJMN"
&RPT$(" ",19)&"$QRKJKJO"&RPT$(" ",20)&"$
 PKJKMN JXXXXXXXXXXXXXXXXXX$ QRJ"
420 S$(3)="KO J^@@@@@@@@@@@@@@@@@@@$\ PJM
NJ^"&RPT$("A",19)&"$_\QR J^"&RPT$("B",18
)&"  $B_\ J^BBBBBJXXXXXXXXXXXX\&%$BB_\"
430 S$(4)="^BBJXXX^F           D_XX$\BB_B
BJ^F"&RPT$(" ",17)&"D$_\BBBJ^"&RPT$(" ",
20)&"$ _\BX^ F press firebutton 1$ D_X"
440 S$(5)=RPT$(" ",23)&"$"&RPT$(" ",32):
: CALL HCHAR(21,1,88,64)
445 !*
450 RESTORE 320 :: FOR I=1 TO 14 :: READ
 A,S$(6):: CALL CHAR(A,S$(6)):: NEXT I
460 S$(6)="' ' ' ' ' '"
470 DISPLAY AT(1,1):S$(1):S$(2):S$(3):S$
(4):S$(5)
480 RESTORE 150 :: FOR I=1 TO 79 :: READ
 A,B,C :: CALL HCHAR(A,B,C):: NEXT I
490 DISPLAY AT(23,2)SIZE(14):"points= 00
000" :: DISPLAY AT(23,18)SIZE(12):"lifes
=/////"
500 DISPLAY AT(24,7)SIZE(17):"highscore=
 00000"
510 CALL COLOR(1,13,15,2,6,15,3,2,15,4,2
,15,5,7,15,6,5,15,7,5,15,8,14,15,9,2,15,
10,2,15,11,2,15)
520 CALL COLOR(12,2,15,13,2,15):: CALL S
CREEN(14):: ON ERROR 1350
525 !*
530 CALL SPRITE(#1,42,7,152,120,0,-3,#2,
46,5,152,128,0,-3,#3,42,7,152,136,0,-3)
540 CALL SPRITE(#4,47,5,24,104,#5,47,7,3
2,128,#6,47,13,40,160,#7,47,11,48,184,#8
,47,16,56,144)
550 CALL SPRITE(#9,47,2,64,96):: F=40
560 FOR I=4 TO 9 :: CALL POSITION(#I,Y,X
):: A=X-120 :: B=Y-40 :: C=-10*SGN(A)+RN
D*4 :: D=-3*SGN(B)+RND*2 :: CALL MOTION(
#I,D,C)
570 CALL PATTERN(#1,F,#3,F):: F=F+1 :: I
F F=43 THEN F=40
580 CALL KEY(1,T,ST):: IF T=18 THEN 600
590 NEXT I :: GOTO 560
```

```
595 !*
600 CALL HCHAR(18,7,32,19)
610 HI=0 :: CALL DELSPRITE(ALL):: DISPLA
Y AT(17,12)SIZE(9):"level";LEV :: CALL S
PRITE(#6,47,2,24,112,#7,47,7,32,152,#8,4
7,11,40,112)
620 CALL SPRITE(#9,47,16,48,152,#10,47,1
3,56,112,#11,47,5,64,152,#12,47,1,184,11
2):: GOSUB 970
630 CALL SPRITE(#2,40,7,152,152,#3,46,5,
152,160,#4,40,7,152,168):: RESTORE 230
635 !*
640 CALL SPRITE(#1,40,2,152,256):: FOR T
=1 TO 3 :: READ A,B,C,D,E :: F=C
650 FOR I=A TO B STEP -2 :: CALL SOUND(-
100,-4,0):: ON E GOTO 660,670,680
660 CALL LOCATE(#1,152,I):: GOTO 690
670 CALL LOCATE(#1,I,200):: GOTO 690
680 CALL LOCATE(#1,96,I)
690 CALL PATTERN(#1,C):: GOSUB 950 :: C=
C+1 :: IF C=D THEN C=F
700 NEXT I :: NEXT T
705 !*
710 CALL PATTERN(#1,68):: CALL SOUND(120
,-7,0):: CALL MOTION(#1,-40,-20)
720 CALL MOTION(#1,15,INT(RND*4)*SGN(RND
-.5)):: CALL PATTERN(#1,69)
730 CALL JOYST(1,X,Y):: IF X=4 THEN 750
ELSE IF X=-4 THEN 770 ELSE CALL MOTION(#
2,0,0,#3,0,0,#4,0,0):: GOSUB 950
740 CALL POSITION(#1,Z,SP):: IF Z>133 TH
EN 790 ELSE 730
750 CALL SOUND(-100,-4,0):: CALL PATTERN
(#2,45,#4,45):: CALL MOTION(#2,0,6,#3,0,
6,#4,0,6):: CALL PATTERN(#2,44,#4,44)::
GOSUB 950
755 !*
760 CALL POSITION(#1,Z,SP):: CALL PATTER
N(#2,43,#4,43):: IF Z>133 THEN 790 ELSE
730
770 CALL SOUND(-100,-4,0):: CALL PATTERN
(#2,42,#4,42):: CALL MOTION(#2,0,-6,#3,0
,-6,#4,0,-6):: CALL PATTERN(#2,41,#4,41)
:: GOSUB 950
775 !*
780 CALL POSITION(#1,Z,SP):: CALL PATTER
N(#2,40,#4,40):: IF Z<133 THEN 730
790 CALL MOTION(#1,0,0):: CALL POSITION(
#1,Z,SP):: CALL LOCATE(#1,148,SP)
800 CALL COINC(#3,#1,7,HIT):: IF HIT THE
N CALL PATTERN(#1,68):: CALL SOUND(140,1
10,30,110,30,800,30,-4,0):: CALL MOTION(
#1,-15,0)ELSE 890
805 !*
810 CALL MOTION(#2,0,0,#3,0,0,#4,0,0)::
FOR B=1 TO 9 :: CALL JOYST(1,X,Y):: CALL
 MOTION(#1,-17,X):: CALL KEY(1,T,ST):: I
F T=18 THEN 830 ELSE GOSUB 950
820 CALL KEY(1,T,ST):: IF T<>18 THEN 880
830 CALL MOTION(#1,0,0):: CALL POSITION(
#1,Z,SP):: ZE=INT(Z/8)+3 :: CALL COINC(#
1,#ZE,12,HIT)
840 CALL MOTION(#1,16,INT(RND*4)*SGN(RND
-.5))
850 IF NOT HIT THEN 720 ELSE CALL PATTER
N(#ZE,67):: CALL PATTERN(#1,69):: CALL S
OUND(300,-6,0):: GOSUB 950 :: CALL DELSP
RITE(#ZE):: HI=HI+1 :: GOSUB 1580
860 IF HI=6 THEN CALL DELSPRITE(#1):: G=
2 :: GOSUB 1160 ELSE 730
870 GOTO 1000
880 NEXT B :: GOTO 720
885 !*
890 F=40 :: CALL DELSPRITE(#3):: CALL SO
UND(300,-7,0):: CALL LOCATE(#1,152,SP)::
 CALL PATTERN(#2,F,#4,F+3):: CALL MOTION
(#2,0,-3,#4,0,5)
900 FOR I=1 TO 15 :: CALL PATTERN(#1,90,
#2,F,#4,F+3):: CALL SOUND(-15,-7,5)
910 GOSUB 950 :: CALL PATTERN(#1,91):: F
=F+1 :: CALL SOUND(-15,-6,5):: IF F=43 T
HEN F=40
920 NEXT I :: CALL DELSPRITE(#2,#4):: FO
R I=1 TO 8 :: FOR B=128 TO 131
930 CALL PATTERN(#1,B):: CALL SOUND(20,-
6,0)
940 GOSUB 950 :: NEXT B :: NEXT I :: CAL
L DELSPRITE(#1):: GOSUB 1590 :: IF LI=5
THEN 1600 ELSE 630
945 !*
950 CALL POSITION(#12,Z,SP):: IF SP<72 T
HEN GOSUB 980 :: RETURN ELSE IF SP>184 T
HEN GOSUB 990
960 RETURN
965 !*
970 FOR I=1 TO 6 :: S(I)=INT(RND*4)+LEV
:: NEXT I :: S(7)=4+LEV
980 CALL MOTION(#12,0,S(7),#6,0,S(1),#7,
0,-S(2),#8,0,S(3),#9,0,-S(4),#10,0,S(5),
#11,0,-S(6)):: RETURN
990 CALL MOTION(#12,0,-S(7),#6,0,-S(1),#
7,0,S(2),#8,0,-S(3),#9,0,S(4),#10,0,-S(5
),#11,0,S(6)):: RETURN
995 !*
1000 HI=0 :: CALL DELSPRITE(ALL):: Z=152
 :: CALL SPRITE(#1,68,2,Z,200):: DISPLAY
 AT(21,5)SIZE(11):S$(6)
1010 FOR I=3 TO 8 :: F=INT(RND*12)+2 ::
CALL SPRITE(#I,47,F,160,I*16):: NEXT I :
: CALL SPRITE(#9,47,1,180,10,0,3)
1020 FOR I=3 TO 8 :: CALL MOTION(#I,INT(
RND*-2)+-1,0):: NEXT I :: CALL SOUND(100
,-2,0)
1025 !*
1030 FOR G=1 TO 140 :: CALL JOYST(1,X,Y)
:: CALL KEY(1,T,ST):: IF T=18 THEN CALL
SPRITE(#2,89,2,Z,192,0,-37):: GOTO 1080
1040 IF Y=4 THEN 1050 ELSE IF Y=-4 THEN
1060 ELSE 1130
1050 Z=Z-4 :: G=G+.7 :: CALL PATTERN(#1,
70):: IF Z<48 THEN Z=48 :: GOTO 1130 ELS
E CALL LOCATE(#1,Z,200):: CALL PATTERN(#
1,71):: GOTO 1130
1060 Z=Z+4 :: G=G+.7 :: CALL PATTERN(#1,
70):: IF Z>152 THEN Z=152 :: GOTO 1130 E
LSE CALL LOCATE(#1,Z,200):: CALL PATTERN
(#1,71)
1070 GOTO 1130
1075 !*
1080 FOR A=1 TO 40 :: CALL COINC(ALL,HIT
):: IF NOT HIT THEN 1120
1090 CALL MOTION(#2,0,0):: CALL COLOR(#2
,1):: FOR B=3 TO 8 :: CALL DISTANCE(#2,#
B,X):: IF X<180 THEN 1110
1100 NEXT B :: CALL DELSPRITE(#2):: GOTO
 1130
1110 CALL PATTERN(#B,67):: G=G+11 :: CAL
L SOUND(100,-5,0):: CALL DELSPRITE(#B,#2
):: HI=HI+1 :: GOSUB 1580 :: IF HI=6 THE
N 1150 ELSE 1130
1120 NEXT A :: CALL DELSPRITE(#2):: G=G+
11
1130 NEXT G :: CALL HCHAR(21,7,88,13)
1140 GOSUB 1590 :: IF LI=5 THEN 1600 ELS
E 1000
1150 CALL HCHAR(21,7,88,14):: CALL DELSP
RITE(ALL):: G=1 :: GOSUB 1160 :: GOTO 12
30
1155 !*
1160 F=3 :: DISPLAY AT(18,6)SIZE(18):"pr
ess firebutton 1"
1170 ON G GOTO 1180,1190
1180 RESTORE 240 :: GOTO 1200
1190 RESTORE 280
1200 READ D :: IF D=99 THEN 1170 ELSE IF
 D<110 THEN READ C ELSE C=D :: D=1
1210 CALL SOUND(INT(210*D),C,0,C-2,0,C+2
,0):: CALL SCREEN(F):: F=F+1 :: IF F=16
THEN F=3
1220 CALL KEY(1,T,ST):: IF T<>18 THEN 12
```

```
00 ELSE CALL HCHAR(18,7,32,19):: HI=0 ::
 CALL SCREEN(14):: RETURN
1225 !*
1230 CALL DELSPRITE(ALL):: R=1 :: V(1)=1
 :: V(2)=2 :: A=15 :: B=-9 :: C=4
1240 CALL SPRITE(#1,134,2,152,96,#2,132,
2,152,160,#3,47,7,148,157)
1250 FOR I=-A TO A STEP 3 :: CALL MOTION
(#3,I,B)
1260 CALL JOYST(1,X,Y):: IF X=-4 THEN 12
70 ELSE IF X=4 THEN 1280 ELSE CALL MOTIO
N(#R,0,0):: GOTO 1290
1270 CALL PATTERN(#R,133):: CALL MOTION(
#R,0,-10):: CALL SOUND(15,-5,8):: CALL P
ATTERN(#R,132):: V(R)=2 :: GOTO 1290
1280 CALL PATTERN(#R,135):: CALL MOTION(
#R,0,10):: CALL SOUND(15,-5,8):: CALL PA
TTERN(#R,134):: V(R)=1
1290 NEXT I :: CALL MOTION(#3,0,0,#R,0,0
):: CALL POSITION(#3,Z,SP,#R,D,E):: CALL
 LOCATE(#3,148,SP)
1300 CALL SOUND(-50,783,15,659,15,523,15
):: CALL COINC(#3,148,E+C,4,HIT):: IF HI
T AND R=V(R)THEN GOSUB 1580 ELSE 1350
1310 HI=HI+1 :: IF HI=6+(LEV*2)THEN G=2
:: GOSUB 1160 :: GOTO 1390
1315 !*
1320 A=INT(RND*(23-15+LEV/2))+15 :: B=IN
T(RND*(11-7+LEV/2))+7 :: ON R GOTO 1330,
1340
1330 R=2 :: C=-4 :: GOTO 1250
1340 B=-B :: R=1 :: C=4 :: GOTO 1250
1345 !*
1350 FOR A=7 TO 1 STEP -1 :: FOR I=-20 T
O 15 STEP 5 :: CALL MOTION(#3,I,B/2):: N
EXT I :: CALL SOUND(5,110*A,A):: NEXT A
:: CALL MOTION(#3,0,0)
1360 CALL SOUND(250,-6,15):: CALL PATTER
N(#3,67):: GOSUB 1590
1365 !*
1370 CALL COLOR(#3,2):: FOR I=1 TO 6 ::
FOR A=128 TO 131 :: CALL PATTERN(#3,A)
1380 CALL SOUND(1,-6,12):: FOR D=1 TO 7
:: NEXT D :: NEXT A :: NEXT I :: CALL DE
LSPRITE(ALL):: IF LI=5 THEN 1600 ELSE 12
30
1385 !*
1390 F=84 :: HI=0 :: CALL DELSPRITE(ALL)
:: CALL SPRITE(#3,F,2,152,255)
1400 FOR A=64 TO 184 STEP 24 :: CALL SPR
ITE(#1,136,7,136,A,#2,137,9,144,A,#4,138
,7,136,A+8,#5,139,9,144,A+8)
1410 CALL SPRITE(#6,68,5,152,A+4,#7,69,7
,118,128)
1415 !*
1420 E=INT(RND*(-17+8+-LEV/2))+-8 :: CAL
L MOTION(#3,0,E)
1430 CALL PATTERN(#3,F):: CALL KEY(1,T,S
T):: IF T=18 THEN 1460
1440 F=F+1 :: IF F=88 THEN F=84
1450 CALL SOUND(100,-4,5):: GOTO 1430
1455 !*
1460 CALL PATTERN(#3,86,#7,68):: CALL MO
TION(#3,-6,E)
1470 FOR D=110 TO 910 STEP 100 :: CALL S
OUND(-10,D,0):: NEXT D :: CALL MOTION(#3
,0,E)
1480 CALL COINC(#3,142,A+8,4,HIT):: CALL
 COINC(#3,#6,24,B):: CALL MOTION(#3,6,E)
1490 FOR D=910 TO 110 STEP -100 :: CALL
SOUND(-10,D,0):: NEXT D :: CALL MOTION(#
3,0,E)
1500 CALL POSITION(#3,Z,SP):: CALL LOCAT
E(#3,152,SP):: CALL PATTERN(#3,85,#7,69)
1510 IF HIT THEN 1570 ELSE IF NOT B THEN
 1430
1515 !*
1520 CALL MOTION(#1,1,-2,#4,1,-2,#2,1,-2
,#5,1,-2):: CALL PATTERN(#6,90):: CALL M
OTION(#3,0,0):: CALL SOUND(200,-6,9)
1530 GOSUB 1590
1540 FOR 1=1 TO 10 :: FOR B=90 TO 91 ::
CALL PATTERN(#6,B,#7,B-22):: CALL SOUND(
20,-7,9):: FOR D=1 TO 15 :: NEXT D :: NE
XT B :: NEXT I
1550 CALL PATTERN(#6,68):: IF LI=5 THEN
1600
1555 !*
1560 CALL MOTION(#1,0,0,#2,0,0,#4,0,0,#5
,0,0):: CALL LOCATE(#1,136,A,#2,144,A,#4
,136,A+8,#5,144,A+8):: GOTO 1420
1570 GOSUB 1580 :: NEXT A :: CALL DELSPR
ITE(ALL):: G=1 :: GOSUB 1160 :: LEV=LEV+
1 :: GOTO 610
1575 !*
1580 PU=PU+(50*LEV):: DISPLAY AT(23,9)SI
ZE(6):PU :: RETURN
1585 !*
1590 G=1 :: LI=LI+1 :: CALL HCHAR(23,26,
83,LI):: RETURN
1600 CALL DELSPRITE(ALL):: DISPLAY AT(16
,10)SIZE(12):"game over in"
1610 IF PU<=HIG THEN 1630
1620 HIG=PU :: DISPLAY AT(24,17)SIZE(6):
HIG
1630 GOSUB 1160 :: CALL HCHAR(16,8,32,16
):: CALL HCHAR(23,26,47,5):: LI,PU=0 ::
LEV=1 :: DISPLAY AT(23,10)SIZE(6):"00000
" :: GOTO 610
1640 END !OKTOBER 1985 (W.D.)
```


Profis schwören auf die ...

GAMES

Mehr darüber finden Sie auf der Seite 2!

IMPRESSUM

Homecomputer 4. Jahrgang
erscheint alle 2 Monate im
Tronic-Verlag, 3444 Wehretal 1
Tel. 05651/40693 oder 40643

Herausgeber: Axel Credé
Chefredakteur: Manfred Kleimann
Redaktion: Siegfried Görk, Hartmut Wendt, Frank Brall, Ottfried Schmidt, Bernd Zimmermann

Programmautoren: V. Becker; J. Berke; H. Born; W.H. Döltsch; K. Ercan; D. Gumtau; M. Hern; T. Lanan; A. Müller; M. Müller; T. Piekarz; M. Sachenbacher; O. Schmidt

Gesamtherstellung:
Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Vertrieb:
Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 06121/2660

Anzeigenverwaltung (Inland):
Hartmut Wendt, Tronic-Verlag, Postfach 41, 3444 Wehretal 1, Tel. 05651/40694

Anzeigenleitung (Ausland):
Public Relation (Foreign Countries):
Manfred Kleimann, c/o Tronic-Verlag, Postfach 41 (Post Box 41)
D-3444 Wehretal 1, Tel.: 05651/40643

Erscheinungsweise: Erstverkaufstag von HOMECOMPUTER ist Anfang des Monats

Anzeigenpreisliste:
Bitte Mediaunterlagen anfordern.

Software-Service und Vertrieb: Annele Kratzenberg, Heike Rabe

Urheberrecht:
Alle in HOMECOMPUTER veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.
Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags.
Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.
Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreise:
Einzelheft 6,-- DM

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen. Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,– DM pro abgedruckter Seite im Heft einverstanden sind.

Titelblatt: Werbeatelier H. Kästle, Eschwege

RAPUNZEL

TI-99/4A - Extended Basic

„Kraft meiner königlichen Gewalt verkünde ich, daß meine Tochter Rapunzel unter allen Umständen im Turm gefangengehalten werden muß!"

Ja, fast wie bei der Original-Vorlage der „Grimm Brothers" geht es darum, die schöne Prinzessin – namens RAPUNZEL – aus dem Turm zu befreien, der ihr schon seit Jahren als Verlies dient. Prinz Rupert – unsterblich in die Schöne verliebt – scheut weder Mühe noch Lebensgefahr, um die Dame seines Herzens da oben rauszuholen. Ein langer und beschwerlicher Weg wartet auf Rupert ...

Dieser lange Weg zum „ewigen Glück" führt bei diesem Spiel durch mehrere Bilder. Prinz Rupert – Sie haben die königliche Ehre, ihn per Joystick (Port 1) zu steuern – wird durch eine Unzahl von Soldaten und Rittern an seinem Vorhaben gehindert. Der Befehl lautet: „Tötet den dreisten Eindringling!".

Doch: Rupert läßt sich nicht beirren, er weicht den Geschossen der Feinde aus und bemüht sich, über eine Schlucht zu gelangen, die den Weg zu Rapunzel „blockiert". Nun steht er da, unser Held, und weiß nicht weiter. Dann kommt ihm die Idee: Er bemerkt eine Wäscheleine, die über der Schlucht hängt. An ihr entdeckt er eine mittelalterliche „Jeans" – Marke „Helm-washed". Die kommt ihm gerade recht. Er springt an sie, und siehe da, ... aber das werdet Ihr ja noch früh genug erfahren! (Manfred Kleimann)

DAS LISTING — **TI-99/4A** — **RAPUNZEL**

```
100 ! ********************        *
 RAPUNZEL       *      *    (C) TIMO PLOKAR
Z  *
110 !*    AM BAHNDAMM 6      *      * 2822
SCHWANEWEDE 2  *        ******************
***
120 SC=0 :: MEN=3 :: RANDOMIZE :: FOR I=
1 TO 14 :: CALL COLOR(I,16,2):: NEXT I :
: CALL COLOR(2,10,2,3,10,2,4,10,2)
130 LEVEL=1
140 DIM V(20)
150 !
160 CALL CLEAR
170 FOR I=1 TO 20 :: V(I)=INT(RND*2)+3 :
: NEXT I
180 CALL SCREEN(2)
190 S1$="000304060422712320C3C34040303030
7008040C040800080404040408080800C0"
200 S2$="207324262422212334 1C0C040303030
7008040C040800080404040408080800C0"
210 MR$(2)="00000000000000000000000000000
00007BC8C6C8D03A445B4B4B303030303030 3B"
220 MR$(1)="0000000000000000000000000000
000078C8C6E8903844 5F4444386C6CC6C6E7"
230 ML$(1)="0000000000000000000000000000
00000E131363110A1C22F222221C363663E7"
240 ML$(2)="0000000000000000000000000000
00001C2E26C6261BA4643424241B1B1B1B1B"
250 CAOS$="73B4B4B7B4740000007 3B4B4641 41
4E36F90909E91EE0000003FA2A2B2A2A222"
260 CALL CHAR(42,"000000000000FFFF",33,"
00DDDDDD00777777")!STONE
270 PF$="000000000000A050FF50A0000000000
00000000000000B0EFF0E0B00000"
280 HO$="1515 1B1F1F1F1F1B111B1F1F0E00000
00404ECFCFC7C7C6C446C7C7C38000"
290 KL$="0000000004072010123 8101A1B07090
0000000008080 4030A020B02040A00"
300 SO$="0103040811111711111101F0606060
E008040201016DA141B1010F0C0C0C0E0"
310 CALL CHAR(40,"FEEFC3FCE77EEFFE")
320 CALL BILD(CAOS$)
```

Sie kennen sicher das Märchen der Gebrüder Grimm? Doch diese Story ist nicht von den „Grimm Brothers" erdacht. Diese ist härter – nichts für schwache Nerven! Es ist der Versuch eines tollkühnen Prinzen, RAPUNZEL – seine Angebetete – aus der Gefangenschaft zu retten. Eine tödliche Mission???

```
330 !
340 ON LEVEL GOTO 350,370,390,420,450,49
0,520,550
350 !BILD 1
360 CALL FELD :: CALL TURM :: CALL SOLDA
T(11,100,S1$,S2$,3):: CALL SOLDAT(3,150,
S1$,S2$,4):: GOTO 760
370 CALL DELSPRITE(ALL):: CALL SOLDAT(16
,170,S1$,S2$,3)
380 CALL SOLDAT(11,100,S1$,S2$,4):: CALL
 KREUSR(9,135,SO$,7):: GOTO 760
390 CALL DELSPRITE(ALL):: CALL PFEIL(PF$
,8,90,25,90):: CALL PFEIL(PF$,9,10,25,90
)
400 CALL SOLDAT(3,100,S1$,S2$,4):: CALL
KREUSR(14,140,SO$,6):: CALL KREUSR(6,180
,SO$,7)
410 GOTO 760
420 CALL DELSPRITE(ALL):: CALL FELD :: C
ALL TURM
430 CALL LOCH(4,S1$,S2$,10):: CALL LOCHK
(12,SO$,15,6):: CALL KLETTE(7,KL$,-17)
440 GOTO 760
450 CALL DELSPRITE(ALL):: CALL FELD :: C
ALL TURM
460 CALL PFEIL(PF$,7,92,21,93):: CALL LO
CHK(3,SO$,10,10):: CALL LOCHK(3,SO$,14,1
1):: CALL LOCHK(3,SO$,18,12)
470 CALL PFEIL(PF$,6,87,37,88)
480 GOTO 760
490 CALL DELSPRITE(ALL):: CALL FELD :: C
ALL TURM
500 CALL PFEIL(PF$,6,01,80,93):: CALL PF
EIL(PF$,7,97,80,88):: CALL LOCHK(9,SO$,1
0,10):: CALL LOCHK(10,SO$,14,11):: CALL
LOCHK(6,SO$,18,12)
510 GOTO 760
520 CALL DELSPRITE(ALL):: CALL FELD :: C
ALL TURM
530 CALL HOSE(HO$)
540 GOTO 760
550 CALL DELSPRITE(ALL):: CALL CLEAR ::
CALL FELD
560 CALL TURM1
570 CALL CHAR(112,"00001C3E7F7F7F7F3F1F0
F0703010000000387CFEFEFEFEFCF8F0E0C08")
580 CALL CHAR(116,"070B09120B0E0B0703030
40B0B0B0B0BC0E0F0F87C5C4E8E0703B14040404
040")
590 CALL SPRITE(#2,116,11,40,174)
600 CALL CHAR(120,"E4AAEECAAAAA000000008
48A8A8E8AEACAAAAACA8A840000000068888442
2CC")
610 CALL CHAR(128,"97D4B4929197000000002
E28E8AEA8EE74447444447700000000A4B4ACA4A
4A4")
620 CALL SPRITE(#3,120,16,60,24):: CALL
SPRITE(#4,128,16,60,40)
630 CALL SPRITE(#5,112,9,40,160,3,-3)
640 CALL CHAR(96,MR$(2)):: CALL SPRITE(#
1,96,8,105,50)
650 DISPLAY AT(11,1)SIZE(10):"  HAAR" ::
 DISPLAY AT(12,1)SIZE(10):"  HERAB"
660 CALL MUSIK :: CALL PFEIL(PF$,6,10,10
,45)
670 FOR I=1 TO 200 :: NEXT I
680 CALL COINC(ALL,C):: IF C THEN 690 EL
SE 680
690 CALL MOTION(#2,0,10)
700 CALL DELSPRITE(#5):: DISPLAY AT(5,24
):"HILFE"
710 FOR I=1 TO 400 :: NEXT I
720 CALL DELSPRITE(#2,#3,#4,#5,#6):: CAL
L HCHAR(11,1,32,10):: CALL HCHAR(12,1,32
,10):: CALL HCHAR(5,24,32,7)
730 DISPLAY AT(11,1)SIZE(10):"   MIST" :
: CALL POSITION(#1,X,Y):: FOR U=1 TO 17
740 CALL LOCATE(#1,X-8,Y):: FOR I=1 TO 2
4 :: NEXT I :: CALL LOCATE(#1,X,Y):: CAL
L SOUND(20,-5,0,110,0):: NEXT U
750 LEVEL=1 :: CALL CLEAR :: CALL DELSPR
ITE(ALL):: GOTO 340
760 CALL CHAR(100,MR$(1),96,MR$(2),92,ML
$(1),88,ML$(2))
770 CALL SPRITE(#1,100,8,105,20)
780 GOSUB 1320
790 RN=RN+1 :: CALL COINC(ALL,C):: IF C
THEN 1020
800 IF RN>18 THEN RN=1
810 CALL POSITION(#1,H,S):: IF S>245 THE
N 1340
820 CALL PATTERN(#1,96):: CALL PATTERN(#
V(RN)+1,116):: CALL JOYST(1,X,Y):: IF X=
0 THEN 890 :: IF X<0 THEN 840 :: CALL PA
TTERN(#1,100)
830 CALL MOTION(#1,0,X*1.5):: CALL PATTE
RN(#V(RN),112):: CALL KEY(1,K,S):: IF S
THEN 920 ELSE 790
840 RN=RN+1 :: CALL COINC(ALL,C):: IF C
THEN 1020
850 IF RN>18 THEN RN=1
860 CALL POSITION(#1,H,S):: IF S<15 THEN
 CALL MOTION(#1,0,16)
870 CALL PATTERN(#1,88):: CALL PATTERN(#
V(RN),116):: CALL JOYST(1,X,Y):: IF X=0
THEN 890 :: IF X>0 THEN 790 :: CALL PATT
ERN(#1,92)
880 CALL MOTION(#1,0,X*1.5):: CALL PATTE
RN(#V(RN),112):: CALL KEY(1,K,S):: IF S
THEN 920 ELSE 840
890 CALL MOTION(#1,0,0):: CALL JOYST(1,X
,Y):: CALL COINC(ALL,C):: IF C THEN 1020
 :: IF X>0 THEN 790 :: IF X<0 THEN 840 :
: RN=RN+1
900 CALL PATTERN(#V(RN),116):: IF RN>15
THEN RN=1
910 CALL PATTERN(#V(RN),112):: GOTO 890
920 CALL SOUND(100,110,0):: CALL POSITIO
N(#1,H,S):: IF X>0 AND S<230 THEN 980
930 IF X<0 AND S>40 THEN 940 ELSE 840
940 CALL MOTION(#1,-11,-8):: FOR I=0 TO
10 :: CALL SOUND(-50,140,I):: CALL COINC
(ALL,C):: IF C THEN 1020
950 NEXT I :: CALL MOTION(#1,11,-8):: FO
R I=10 TO 0 STEP -1 :: CALL SOUND(-50,15
0,I):: CALL COINC(ALL,C):: IF C THEN 102
0
960 NEXT I
970 CALL MOTION(#1,0,0):: CALL POSITION(
#1,F,S):: CALL LOCATE(#1,105,S):: GOSUB
1320 :: GOTO 870
980 CALL MOTION(#1,-11,8):: FOR I=0 TO 1
0 :: CALL SOUND(-50,150,I):: CALL COINC(
ALL,C):: IF C THEN 1020
990 NEXT I :: CALL MOTION(#1,11,8):: FOR
 I=10 TO 0 STEP -1 :: CALL SOUND(-50,140
,I):: CALL COINC(ALL,C):: IF C THEN 1020
1000 NEXT I
1010 CALL MOTION(#1,0,0):: CALL POSITION
(#1,F,S):: CALL LOCATE(#1,105,S):: GOSUB
 1320 :: GOTO 870
1020 CALL COINC(#1,#2,14,C):: IF C<>0 TH
EN 1040
1030 GOTO 1090
1040 CALL MOTION(#1,0,4)
1050 CALL POSITION(#1,H,S):: IF S>160 TH
1060 SC=SC+7 :: CALL SOUND(100,330,0)::
CALL SOUND(140,550,0):: CALL SOUND(300,6
60,0):: CALL SOUND(500,330,0)
1070 GOTO 1050
1080 CALL LOCATE(#1,105,S):: CALL DELSPR
ITE(#2):: GOSUB 1320 :: CALL KLETTE(10,K
L$,-10):: GOTO 790
1090 MEN=MEN-1 :: IF MEN=0 THEN 1120
1100 CALL COLOR(#1,5):: CALL COLOR(#1,3)
:: CALL COLOR(#1,6)
1110 CALL MOTION(#1,80,0):: FOR I=1 TO 3
0 STEP 5 :: CALL SOUND(-200,-5,I):: NEXT
 I :: CALL DELSPRITE(#1):: GOTO 340
1120 CALL DELSPRITE(ALL):: CALL CLEAR
```

```
1130 CALL CHAR(35,"FFFFFFFFFFFFFFFF",36,
"7E7E7E7E7E7E7E7E",37,"3C3C3C3C3C3C3C3C"
,38,"1818181818181818")
1140 FOR I=1 TO 13 :: CALL COLOR(I,9,2):
: NEXT I :: CALL SCREEN(2)
1150 PRINT "   ####  ####  #   #  ####
   &     &  &  &# #&  &        % ##  ###
#  % # %  ###  "
1160 PRINT "   $  $  $  $  $   $  %
   ####  #  #  #   #  #### "
1170 PRINT : :
1180 PRINT "   ####  &  &  ####  ####
   &  &  %  %  &     &  &      %  %  $
$  ###   ####      $  $  #  #  %     % #
   "
1190 PRINT "   ####   ##   ####  $  #  "
1200 PRINT : : : :
1210 IF SC>HSC THEN HSC=SC
1220 GOSUB 1320
1230 DISPLAY AT(22,1):"   PRESS BEGIN TO
 START "
1240 CALL SOUND(-100,440,0):: CALL SOUND
(-110,520,0):: CALL SOUND(-80,440,1):: C
ALL SOUND(-220,220,0):: CALL SOUND(-200,
520,0)
1250 FOR I=1 TO 200 :: NEXT I
1260 CALL KEY(0,K,S):: IF K=14 THEN 1300
1270 CALL SOUND(-2,RND*110+110,RND*2)::
GOTO 1260
1280 GOTO 1260
1290 !
1300 FOR I=1 TO 14 :: CALL COLOR(I,16,2)
:: NEXT I :: CALL COLOR(2,10,2):: LEVEL=
1 :: MEN=3 :: SC=0
1310 CALL CLEAR :: GOTO 330
1320 DISPLAY AT(1,1):"SCORE ";SC;"     M
EN ";MEN;:: DISPLAY AT(2,1):"HI SCORE ";
HSC
1330 RETURN
1340 SC=SC+(110+LEVEL):: LEVEL=LEVEL+1 :
: GOTO 340
1350 SUB MUSIK
1360 FOR S1=1 TO 27 :: READ D,E :: CALL
SOUND(D,E,0):: NEXT S1
1370 DATA 150,294,150,330,150,349,150,39
2,150,440,150,440,300,440,150,523,150,46
6,150,440,150,392
1380 DATA 300,440,300,440,150,466,150,39
2,150,392,150,392,150,440,150,349,150,34
9,150,349,150,440
1390 DATA 150,392,150,349,150,330,300,29
4,300,294
1400 RESTORE 1360
1410 SUBEND
1420 SUB TURM1
1430 DISPLAY AT(7,1):"
 ?,?,?,-                    .    /
                 :    :
     : ; ;:"
1440 DISPLAY AT(11,1):"
  :    :                    :    :
                 :    :"
1450 DISPLAY AT(14,1):"
  :    :                    :    :"
1460 SUBEND
1470 SUB TURM
1480 CALL CHAR(63,"FF8080808080808080",44,
"8080808080808080FF",45,"FF8181818181818181"
,46,"8040201008040201")
1490 CALL CHAR(47,"0102040810204080",58,
"0101010101010101",59,"00182442818181FF"
,60,"0101017171710101")
1500 CALL CHAR(61,"FF01010101010101",62,
"FF")
1510 CALL HCHAR(3,1,62,32)
1520 DISPLAY AT(4,1):"      ?,?,-     ?,
?,-          .   /     .   /
   :  :>>>>>>=  :  "
1530 DISPLAY AT(7,1):"      : ;:      :
;:              :  :  />. :  :
   :  : :  < :  :"
1540 DISPLAY AT(10,1):"      :  : :  : :
  :              >>>>>>>>>>>>> "
1550 SUBEND
1560 SUB HOSE(HO$)
1570 CALL CHAR(112,"FFFFFFFFFFFFFFFF0000
000000000000FFFFFFFFFFFFFFFF000000000000
")
1580 FOR I=16 TO 21 :: CALL HCHAR(I,8,32
,8):: NEXT I
1590 CALL HCHAR(12,1,62,32):: CALL CHAR(
104,HO$)
1600 CALL SPRITE(#2,104,5,90,1,0,4)
1610 CALL SPRITE(#6,112,1,117,59):: CALL
 SPRITE(#7,112,1,117,82):: CALL SPRITE(#
8,112,1,117,105)
1620 SUBEND
1630 SUB KREUSR(COL,SEITE,SO$,SPR)
1640 CALL CHAR(120,SO$):: CALL SPRITE(#S
PR,120,COL,107,SEITE)
1650 SUBEND
1660 SUB LOCH(COL,S1$,S2$,E)
1670 CALL HCHAR(16,E,32,3):: CALL HCHAR(
17,E,42,3)
1680 CALL CHAR(112,S1$,116,S2$):: CALL S
PRITE(#3,112,COL,119,E*8-7)
1690 CALL SPRITE(#4,112,COL,119,E*8+5)
1700 SUBEND
1710 SUB LOCHK(COL,SO$,E,SPR)
1720 CALL HCHAR(16,E,32,2):: CALL HCHAR(
17,E,42,2)
1730 CALL CHAR(132,SO$):: CALL SPRITE(#S
PR,132,COL,118,E*8-7)
1740 SUBEND
1750 SUB KLETTE(SPR,KL$,G)
1760 CALL CHAR(128,KL$):: CALL SPRITE(#S
PR,128,15,100,10,0,G)
1770 SUBEND
1780 SUB PFEIL(PF$,SPR,ENTF,G,H)
1790 CALL CHAR(124,PF$):: CALL SPRITE(#S
PR,124,10,H,ENTF,0,G)
1800 SUBEND
1810 SUB BILD(CAOS$)
1820 CALL MAGNIFY(3):: CALL CHAR(100,CAO
S$):: CALL SPRITE(#1,100,15,160,113)
1830 CALL TURM :: DISPLAY AT(13,1):"
   (((((((((((                  (
  (                 ( RAPUNZEL (        "
1840 DISPLAY AT(16,1):"        (    BY
 (                 (  APOLLO  (
      (            (                  ((((((
(((((("
1850 CALL MUSIK
1860 DISPLAY AT(24,1):"   PRESS ANY KEY
TO BEGIN"
1870 CALL SOUND(100,-8,24):: CALL KEY(0,
K,S):: CALL COLOR(#1,RND*14+2):: IF S TH
EN 1880 ELSE 1870
1880 FOR I=1 TO 14 :: CALL COLOR(I,16,2)
:: NEXT I
1890 SUBEND
1900 SUB SOLDAT(COL,MANN,S1$,S2$,SPR)
1910 CALL CHAR(112,S1$,116,S2$)
1920 CALL SPRITE(#SPR,112,COL,106,MANN)
1930 SUBEND
1940 SUB FELD
1950 CALL COLOR(2,15,2):: CALL COLOR(1,1
5,2):: CALL CLEAR :: CALL DELSPRITE(ALL)
1960 CALL HCHAR(16,1,33,32*9)
1970 SUBEND
```

RAPUNZEL · RAPUNZEL · RAPUNZEL · RAPUNZEL · RAPUNZEL · RAPUNZEL

Nur Mut, Rupert! Du wirst es schon schaffen!

DARK LEGION

Ein tolles Weltraumspiel auf dem TI-99

Wieder einmal bedrohen feindliche Mächte Ihren Lebensraum. Kämpfen Sie gegen die „Dark Legion" um das Weiterbestehen der Erde!

Superspiel ✦ Superspiel ✦ Superspiel

Im Jahr 2069 haben sich auf der Erde die einzelnen Staaten endlich zu einer gemeinsamen Weltnation zusammengeschlossen. Nachdem diese große Union den Menschen allmählich Wohlstand und Fortschritt bringt, droht der Weltbevölkerung plötzlich eine tödliche Gefahr. Die schwarzen Legionen des feindlich gesinnten Planeten Kasiopeia sind im Anflug auf die Erde. Sie werden nun vom obersten Weltsicherheitsrat beauftragt, sich mit der Besatzung Ihres Raumschiffes dem übermächtigen Feind entgegenzuwerfen. Zum Wohle der Menschheit und für das Weiterbestehen der Erde stürzen Sie sich sofort in die risikoreiche Mission. (S.G.)

Dark Legion beeindruckt durch seine dreidimensionale Farbgrafik und die hohe Spielgeschwindigkeit.

Als im Jahre 2069 die für undurchdringbar gehaltene Kraftfeldbarriere der Erde unverhofft durch Legionen des feindlichen Machtblocks Kasiopeia zerstört wird, erreichen die Auseinandersetzungen um den Anspruch auf die sagenumwobene Welt Antigra ihren Höhepunkt... In diesem vielleicht letzten Kampf liegt es an Ihnen, das Schicksal der Erde zu dirigieren! Mit einem laserbestückten Raumjäger verteidigen Sie sich gegen eine Flotte fremder Kampfmaschinen, die in rasender Geschwindigkeit über Ihre Festung donnern. Sturzflug! Zielen! - Feuer! Nur ein gekonnter Pilot sammelt hier Punkte! Doch das Vordringen des Feindes scheint unaufhaltsam. Schon zerstören Panzer die Festungsmauern und bahnen sich ihren Weg zum Zentrum. Nach einem Bildwechsel befinden Sie sich nun im Inneren der Festung und feuern aus allen Rohren, um die für Sie tödliche Landung Ihres Kontrahenten zu verhindern. Achten Sie besonders auf die zu Beginn einer jeden Angriffswelle abgefeuerten Peiltorpedos (abschießen!)!

Alle anderen Informationen, z.B. Steuerung etc. werden im Programm erläutert.

ACHTUNG! „Fertig machen" zum Eintippen!

```
10 CALL CLEAR
20 LIFES=4
30 CALL CHAR(48,"7E42466A72627E")
40 CALL CHAR(49,"3808081818183C")
50 CALL CHAR(50,"7E42027E60607E")
60 CALL CHAR(51,"3E02021E06067E")
70 CALL CHAR(52,"404044447E0C0C")
80 CALL CHAR(53,"3E20203E06467E")
90 CALL CHAR(54,"7E40407E62627E")
100 CALL CHAR(55,"7E420408181818")
110 CALL CHAR(56,"3C24247E62627E")
120 CALL CHAR(154,"00001")
130 CALL CHAR(57,"7E42427E06067E")
140 FOR Z=65 TO 87
150 READ CH$
160 CALL CHAR(Z,CH$)
170 NEXT Z
180 DATA "3C247E62626262","7C243E3232327
E","7E42406060627E","7E22223232327E","7E
40407860607E"
190 DATA "7E404078606060","7E42406E62627
E"
200 DATA "4242427E626262","1010101818181
8","0202020606463C","4244487E626262","40
40406060607E"
210 DATA "665A4262626262","724A4A6A6A6A6
6","7E42426262627E","7E42427E606060","7E
4242626A647A"
220 DATA "7E42427E646262","7E42407E06467
E","7C101018181818","4242426262627E","42
42422424181818","6262626425A66"
230 FOR COL=1 TO 16
240 CALL COLOR(COL,16,1)
250 NEXT COL
260 CALL SCREEN(2)
270 CALL CLEAR
280 CALL CHAR(96,"FBFBFB00DFDFDF")
290 CALL CHAR(104,"FFFFFFFFFFFFFFFF")
300 CALL CHAR(105,"0103070F1F3F7FFF")
310 CALL CHAR(112,"0103070F1F3F7FFF")
320 CALL CHAR(121,"0103070F1F3F7FFF")
330 CALL CHAR(122,"FFFFFFFFFFFFFFFF")
340 CALL CHAR(106,"FFFEFCF8F0E0C080")
350 CALL CHAR(94,"0103070F1F3F7FFF")
360 CALL CHAR(36,"4242423C181818")
370 CALL CHAR(37,"4224187C626262")
```

```
380 CALL COLOR(8,8,4)
390 CALL COLOR(9,4,1)
400 CALL COLOR(10,5,8)
410 CALL COLOR(11,1,5)
420 CALL COLOR(12,8,1)
430 CALL COLOR(16,16,1)
440 PRINT "              T H E":::"
  D A R K":::"           L E G I O N":::::
450 PRINT "        yi"
460 PRINT "       yihi"
470 PRINT "       'hhp"
480 PRINT "  yi yzip"
490 PRINT " yihyziji                    yzz"
500 PRINT " 'hhi'jip                   yi'''"
510 PRINT " 'hhh''p                   yip"
520 PRINT " 'hhp                     yzzzzz"
530 PRINT "y'hp          yzzzzzi     ''''''"
540 PRINT "''p           ''''''jzi
          ''''''''p"
550 H=16
560 L=18
570 GOSUB 4620
580 L=15
590 H=18
600 GOSUB 4700
610 GOSUB 4770
630 CALL CHAR(159,"80E0F83FF8E080")
640 GOSUB 4170
650 RANDOMIZE
660 FOR TIME=1 TO 500
670 NEXT TIME
680 CALL CLEAR
690 PRINT "THE DARK LEGION HAS BROKEN"::
700 PRINT "THROUGH THE EARTH'S FORCE"::
710 PRINT "SHIELD AND WANTS TO KILL ALL"
::
720 PRINT "LIFE !!!"
730 GOSUB 4460
740 H=18
750 L=24
760 GOSUB 4620
770 GOSUB 4770
780 GOSUB 4850
790 CALL CLEAR
800 PRINT "$OU ARE THE COMMANDER OF THE"
::
810 PRINT "FORTRESS DEFENCE S$STEMS AND"
::
820 PRINT "CONTROL A PLASMA CANNON.":::
830 PRINT "TR$ TO SHOOT DOWN THE ALIENS"
::
840 PRINT "AND SAVE OUR PLANET !!!":::
850 PRINT ::"K E $ B O A R D :":::
860 PRINT "U P        =    'E'"::
870 PRINT "D O W N    =    '%'"::
880 PRINT "F I R E    =    'SPACE'":::
890 PRINT "PRESS AN$ KE$ TO CONTINUE !"
900 CALL KEY(0,K,S)
910 CALL SOUND(-2,INT(RND*3000+110),0)
920 IF S=0 THEN 900
930 CALL CLEAR
940 CALL SCREEN(2)
950 SCO=0
960 FOR COL=2 TO 7
970 CALL COLOR(COL,2,8)
980 NEXT COL
990 GOSUB 4020
1000 CALL CHAR(156,"814234383C344281")
1010 CALL CHAR(155,"00000000AA")
1020 CALL CHAR(157,"0001071FFE1F0701")
1030 CALL HCHAR(13,25,157)
1040 GOSUB 3920
1050 Y=13
1070 FOR ATTACK=1 TO 20
1080 GOSUB 3530
1090 YI=INT(RND*4+9)
1100 CALL SCREEN(10)
1110 CALL SOUND(-300,-5,0)
1120 CALL SCREEN(2)
1130 GOSUB 3450
1140 GOSUB 3640
1150 FOR XI=8 TO 32 STEP 4
1160 CALL HCHAR(YI,XI,159)
1170 CALL KEY(0,K,S)
1180 IF S=0 THEN 1200
1190 GOSUB 3170
1200 CALL HCHAR(YI,XI,32)
1210 IF HIT=1 THEN 1230
1220 NEXT XI
1230 HIT=0
1240 NEXT ATTACK
1260 CALL CHAR(123,"FCFCFFFCF0F0F8FF")
1270 CALL CHAR(124,"1F001F1F000001FF")
1280 FOR PX=4 TO 10
1290 CALL SOUND(-400,-6,9,110,5)
1300 CALL HCHAR(18,PX-1,122,2)
1310 CALL HCHAR(18,PX,123)
1320 CALL HCHAR(18,PX+1,124)
1330 TIME=SIN(PX)
1340 NEXT PX
1350 CALL CHAR(126,"FFFFFFF3E3C3FFFF")
1360 CALL CHAR(125,"FFFFE7E7FFFFFFFF")
1390 LIFES=4
1400 GOSUB 4980
1410 CALL HCHAR(18,19,126)
1420 GOSUB 4980
1430 CALL VCHAR(18,19,122,2)
1440 CALL HCHAR(18,20,94)
1450 GOSUB 4980
1460 CALL HCHAR(18,20,122)
1470 GOSUB 4980
1480 CALL HCHAR(19,20,122,2)
1490 GOSUB 4980
1500 CALL HCHAR(15,24,96,2)
1510 CALL HCHAR(15,23,32)
1520 TIME=SIN(PX)
1530 FOR PX=10 TO 20
1540 CALL SOUND(-2400,-6,9,110,5)
1550 CALL HCHAR(18,PX-1,122,2)
1560 CALL HCHAR(18,PX,123)
1570 CALL HCHAR(18,PX+1,124)
1580 TIME=SIN(PX)
1590 NEXT PX
1600 CALL HCHAR(18,20,122)
1610 CALL HCHAR(18,21,123)
1620 TIME=SIN(PX)
1630 CALL HCHAR(18,21,122)
1640 FOR SOUND=9 TO 30 STEP .5
1650 CALL SOUND(-400,-6,SOUND,110,SOUND-
4)
1660 NEXT SOUND
1680 CALL SOUND(-4200,-4,0)
1690 CALL HCHAR(23,3,122,22)
1700 FOR COL=1 TO 7
1710 CALL COLOR(COL,16,1)
1720 NEXT COL
1730 P$="    W A R N I N G  !!!"
1740 PX=3
1750 PY=1
1760 GOSUB 4920
1770 CALL SOUND(-4250,-4,0)
1780 P$="THE ALIENS TR$ TO OVERRUN"
1790 PY=5
1800 GOSUB 4920
1810 CALL SOUND(-2000,-4,0)
1820 P$="    $OUR FORTRESS  !!!"
1830 PY=7
1840 GOSUB 4920
1850 GOSUB 4850
1870 FOR TIME=1 TO 3
1880 FOR BX=4 TO 20
1890 CALL SOUND(-800,-6,9,110,5)
1900 CALL HCHAR(18,BX-1,122,2)
1910 CALL HCHAR(18,BX,123)
1920 CALL HCHAR(18,BX+1,124)
1930 TIM=SIN(BX)
1940 NEXT BX
1950 CALL HCHAR(18,20,122)
1960 CALL HCHAR(18,21,123)
1970 TIM=SIN(BX)
```

```
1980 CALL HCHAR(18,21,122)
1990 NEXT TIME
2000 FOR TIME=1 TO 400
2010 NEXT TIME
2020 CALL CHAR(136,"0001071FFE1F0701")
2030 CALL CHAR(137,"0000000011")
2040 CALL CHAR(138,"004224381C244200")
2050 CALL CLEAR
2060 GOSUB 5120
2070 GOSUB 3710
2080 PX=2
2090 PY=21
2100 P$="SCOREz"&STR$(SCO)
2110 GOSUB 5070
2120 CALL HCHAR(1,1,32,320)
2130 FOR COL=2 TO 7
2140 CALL COLOR(COL,2,8)
2150 NEXT COL
2170 GOSUB 4920
2180 XG=15
2190 X=15
2200 CALL CHAR(126,"FFEFEFEFEF0101BD")
2210 CALL HCHAR(15,X,126)
2220 CALL CHAR(152,"1010101010101010")
2230 CALL CHAR(136,"183C5AFF18244281")
2240 CALL CHAR(137,"18FF99FF18244281")
2250 CALL CHAR(138,"2828383838381010")
2260 W1=0
2270 GOTO 5500
2280 W2=0
2290 W3=0
2300 X1=INT(RND*15)+5
2310 X2=INT(RND*5)+3+X1
2320 X3=INT(RND*5)+3+X2
2330 FOR YI=2 TO 14
2340 IF YI<>6 THEN 2360
2350 GOSUB 3550
2360 XA1=X1
2370 XA2=X2
2380 XA3=X3
2390 IF RND*2<.8 THEN 2440
2400 X1=X1-1
2410 X2=X2-1
2420 X3=X3-1
2430 GOTO 2470
2440 X1=X1+1
2450 X2=X2+1
2460 X3=X3+1
2470 IF X1<9 THEN 2440
2480 IF X3>25 THEN 2400
2490 IF W1=1 THEN 2520
2500 CALL HCHAR(YI-1,XA1,32)
2510 CALL HCHAR(YI,X1,136)
2520 IF W2=2 THEN 2550
2530 CALL HCHAR(YI-1,XA2,32)
2540 CALL HCHAR(YI,X2,136)
2550 IF W3=3 THEN 2580
2560 CALL HCHAR(YI-1,XA3,32)
2570 CALL HCHAR(YI,X3,137)
2580 CALL HCHAR(15,XG,122)
2590 CALL HCHAR(15,X,126)
2600 XG=X
2610 CALL KEY(0,KEY,STATUS)
2620 IF STATUS=0 THEN 2790
2630 IF KEY<>83 THEN 2670
2640 IF X=8 THEN 2790
2650 X=X-1
2660 GOTO 2790
2670 IF KEY<>68 THEN 2710
2680 IF X=27 THEN 2790
2690 X=X+1
2700 GOTO 2790
2710 IF KEY<>32 THEN 2790
2720 CALL SOUND(-90,4000,0)
2730 CALL VCHAR(1,X,152,14)
2740 IF X1=X THEN 2920
2750 IF X2=X THEN 2960
2760 IF X3=X THEN 3000
2770 CALL VCHAR(1,X,32,14)
2780 REM
2790 T=T+1
2800 IF T<>2 THEN 2580
2810 T=0
2820 NEXT YI
2830 CALL HCHAR(14,8,32,23)
2840 CALL SCREEN(16)
2850 CALL SOUND(-400,-5,0)
2860 CALL SCREEN(2)
2870 GOSUB 5180
2880 FOR SOUND=110 TO 2000 STEP 50
2890 CALL SOUND(-1,SOUND,0)
2900 NEXT SOUND
2910 GOTO 2260
2920 XW=X1
2930 IF W1=1 THEN 2770
2940 W1=1
2950 GOTO 3030
2960 XW=X2
2970 IF W2=2 THEN 2770
2980 W2=2
2990 GOTO 3030
3000 XW=X3
3010 IF W3=3 THEN 2770
3020 W3=3
3030 CALL VCHAR(1,XW,32,14)
3040 CALL HCHAR(YI,XW,156)
3050 CALL SOUND(-300,-5,0)
3060 SCO=SCO+50
3070 CALL HCHAR(YI,XW,32)
3080 P$=STR$(SCO)
3090 PX=8
3100 PY=21
3110 GOSUB 4920
3120 B=W1+W2+W3
3130 IF B=6 THEN 2260
3140 GOTO 2810
3150 GOTO 3150
3170 IF K<>32 THEN 3330
3180 CALL HCHAR(Y,1,155,24)
3190 CALL SOUND(-50,4000,0)
3200 CALL HCHAR(Y,1,32,24)
3210 IF XI>25 THEN 3430
3220 IF YI<>Y THEN 3430
3230 CALL HCHAR(YI,XI,156)
3240 CALL SOUND(-300,-5,0)
3250 CALL HCHAR(YI,XI,32)
3260 HIT=1
3270 SCO=SCO+10
3280 PX=9
3290 PY=23
3300 P$=STR$(SCO)
3310 GOSUB 4920
3320 RETURN
3330 CALL HCHAR(Y,25,32)
3340 IF K<>69 THEN 3390
3350 IF Y=4 THEN 3370
3360 Y=Y-1
3370 CALL HCHAR(Y,25,157)
3380 RETURN
3390 IF K<>88 THEN 3430
3400 IF Y=13 THEN 3420
3410 Y=Y+1
3420 CALL HCHAR(Y,25,157)
3430 RETURN
3450 CALL HCHAR(YI-4,3,159)
3460 CALL HCHAR(YI-4,3,32)
3470 CALL HCHAR(YI-3,5,159)
3480 CALL HCHAR(YI-3,5,32)
3490 CALL HCHAR(YI-2,7,159)
3500 CALL HCHAR(YI-2,7,32)
3510 RETURN
3530 IF INT(RND*5)=3 THEN 3550
3540 RETURN
3550 CALL CHAR(153,"1020408080402010")
3560 LX=INT(RND*8+10)
3570 CALL VCHAR(1,LX,153,12)
3580 CALL SOUND(-300,-5,2)
3590 CALL SOUND(-300,-6,0)
3600 CALL SOUND(-300,-5,2)
3610 CALL VCHAR(1,LX,32,12)
```

```
3620 RETURN
3640 R=INT(RND*3)
3650 IF R<>1 THEN 3680
3660 CALL CHAR(159,"0000F0FFF")
3670 RETURN
3680 CALL CHAR(159,"80E0F83FF8E080")
3690 RETURN
3710 PRINT "   STOP THE ALIENS  !!!"::
3720 PRINT "LEFT   =  'S'
RIGHT  =  'D'                FIRE   =  'S
PACE'"::
3730 PRINT "IF THE$ REACH $OUR FORTRESS
    $OU LOSE A LIFE !!!"
3740 PRINT "zi":"jzzi":"""h"
3750 PRINT """h"
3760 PRINT "zi''jzzzzzzzzzzzzzzzzzzzzzz
*
3770 PRINT "'hzzzzzzzzzzzzzzzzzzzzzzzzz
*
3780 PRINT "'hzzzzzzzzzzzzzzzzzzzzzzzzi
*
3790 PRINT "'jzzzzzzzzzzzzzzzzzzzzzz''j
*
3800 PRINT "''''''''''''''''''''''''''''
*
3810 PRINT "zzzzzzzzzzzzzzzzzzzzi''''''
*
3820 PRINT "zzzzzzzzzzzzzzzzzzzzih''''''
*
3830 PRINT "zzzzzzzzzzzzzzzzzzzzihh''''''
*
3840 PRINT "zzzzzzzzzzzzzzzzzzzzihhh''''''
*
3850 FOR TIME=15 TO 21 STEP 2
3860 CALL HCHAR(20,TIME,144)
3870 CALL HCHAR(21,TIME,145)
3880 CALL HCHAR(22,TIME,146)
3890 NEXT TIME
3900 RETURN
3920 FOR TIME=1 TO 2
3930 FOR SOU=110 TO 180 STEP 4
3940 CALL SOUND(-300,SOU,TIME,SOU+1,TIME
,SOU+2,TIME)
3950 NEXT SOU
3960 FOR SOU=180 TO 110 STEP -4
3970 CALL SOUND(-300,SOU,TIME,SOU+1,TIME
,SOU+2,TIME)
3980 NEXT SOU
3990 NEXT TIME
4000 RETURN
4020 PRINT "                        yzzzzzz
*
4030 PRINT "                       yzi'''''
*
4040 PRINT "zzzzi                 yzzzzzzzz
*
4050 PRINT "''''jzzzzzzzzzzzzzz''''''''''
*
4060 PRINT "zzzzzzzzzzzzzzzzzz''''''''''
*
4070 PRINT "zzzi'''''^zzzzzz''''''''''''
*
4080 PRINT "''''jzzzzz''''''''''''''''''
*
4090 PRINT "''''''''''''''''''''''''''''
*
4100 PRINT "zzzzzzzzzzzzzzzzzzzzzzzzi''''
*
4110 PRINT "SCOREzzzOzzzzzzzzzzzzzzih''''
*
4120 CALL HCHAR(3,17,154)
4130 CALL HCHAR(4,12,154)
4140 CALL HCHAR(15,15,154)
4150 RETURN
4160 PY=24
4170 REM  -- SPACESHIPS --
4180 REM  ---  COMING  ---
4190 REM
4200 FOR TIME=1 TO 5
4210 R=INT(RND*3+1)
4220 CALL SCREEN(10)
4230 CALL SOUND(-400,-5,0)
4240 CALL SCREEN(2)
4250 IF R<>1 THEN 4330
4260 CALL HCHAR(3,3,159)
4270 CALL HCHAR(3,3,32)
4280 CALL HCHAR(5,5,159)
4290 CALL HCHAR(5,5,32)
4300 CALL HCHAR(7,7,159)
4310 CALL HCHAR(7,7,32)
4320 GOTO 4400
4330 IF R<>2 THEN 4440
4340 CALL HCHAR(16,3,159)
4350 CALL HCHAR(16,3,32)
4360 CALL HCHAR(14,5,159)
4370 CALL HCHAR(14,5,32)
4380 CALL HCHAR(12,7,159)
4390 CALL HCHAR(12,7,32)
4400 FOR X=9 TO 32 STEP 4
4410 CALL HCHAR(10,X,159)
4420 CALL HCHAR(10,X,32)
4430 NEXT X
4440 NEXT TIME
4450 RETURN
4460 REM  -- BASE 1 --------
4470 PRINT
4480 PRINT "         yzzzzzzzi"
4490 PRINT "        yzi'''^ziji"
4500 PRINT "       yzih''^zijih"
4510 PRINT "      yzihh'^zijihh"
4520 PRINT "     yzihhh^zijijii"
4530 PRINT "    yzihhhjzijijihp"
4540 PRINT "   yzzzzzzzijijiji"
4550 PRINT " y''''''''jih'h'p"
4560 PRINT " ''''''''''hjip"
4570 PRINT " ''^zzzi'''h'p"
4580 PRINT " '^''''ji''p"
4590 PRINT "   '^zi''p"
4600 PRINT "    ''p"
4610 RETURN
4620 REM  --BIG ALIEN--
4630 CALL COLOR(13,1,1)
4640 CALL CHAR(128,"040F1F3F217FFEFC")
4650 CALL CHAR(129,"20F0F8FC84FE7F3F")
4660 CALL CHAR(130,"0F3F7080404")
4670 CALL CHAR(131,"F987")
4680 CALL CHAR(132,"9FE1")
4690 CALL CHAR(133,"F0FC0E010202")
4700 CALL HCHAR(H,L,128)
4710 CALL HCHAR(H,L+1,129)
4720 CALL HCHAR(H+1,L-1,130)
4730 CALL HCHAR(H+1,L,131)
4740 CALL HCHAR(H+1,L+1,132)
4750 CALL HCHAR(H+1,L+2,133)
4760 RETURN
4770 REM  ---EFFECT 1--
4780 FOR TIME=1 TO 3
4790 FOR TIII=1 TO 16
4800 CALL COLOR(13,TIII,1)
4810 CALL SOUND(-200,-7,0)
4820 NEXT TIII
4830 NEXT TIME
4840 RETURN
4850 REM  --SCREEN EFFECT-
4860 FOR SCR=1 TO 5
4870 FOR SC=16 TO 2 STEP -1
4880 CALL SCREEN(SC)
4890 NEXT SC
4900 NEXT SCR
4910 RETURN
4920 REM  --PRINT-ROUTINE
4930 FOR PRI=1 TO LEN(P$)
4940 CODE=ASC(SEG$(P$,PRI,1))
4950 CALL HCHAR(PY,PX+PRI,CODE)
4960 NEXT PRI
4970 RETURN
4980 REM ---FIRE PZ
4990 FOR PX=12 TO 18 STEP 2
5000 CALL HCHAR(18,PX,125)
```

```
5010 CALL HCHAR(18,PX,122)
5020 NEXT PX
5030 CALL SOUND(-500,-5,0)
5040 CALL SCREEN(9)
5050 CALL SCREEN(2)
5060 RETURN
5070 REM
5080 FOR SOUND=1 TO 160
5090 CALL SOUND(-2,INT(RND*3000+110),0
5100 NEXT SOUND
5110 RETURN
5120 REM - MAN
5130 CALL COLOR(15,2,8)
5140 CALL CHAR(144,"003C5A7E24187EBD")
5150 CALL CHAR(145,"BDBDBDBDBD993C3C")
5160 CALL CHAR(146,"2424242424242466")
5170 RETURN
5180 REM  - MAN-1
5190 LIFES=LIFES-1
5200 IF LIFES<0 THEN 5230
5210 CALL VCHAR(20,LIFES*2+15,122,3)
5220 RETURN
5230 CALL HCHAR(21,3,122,14)
5240 FOR COL=2 TO 7
5250 CALL COLOR(COL,16,1)
5260 NEXT COL
5270 REM - GAME OVER
5280 CALL SOUND(-2800,-5,0)
5290 GOSUB 4850
5300 CALL SOUND(-4000,-4,0)
5310 PY=1
5320 PX=3
5330 P$="     G A M E    O V E R"
5340 GOSUB 4920
5350 PY=4
5360 P$="          SCORE "&STR$(SCO)
5370 CALL SOUND(-4000,-4,0)
5380 GOSUB 4920
5390 PY=7
5400 P$="   PLA$ AGAIN ? ( $ / N )"
5410 CALL SOUND(-4000,-4,0)
5420 GOSUB 4920
5430 CALL KEY(0,K,S)
5440 CALL SOUND(-999,-8,0)
5450 SOUND=SOUND+10
5460 IF S=0 THEN 5430
5470 IF K=89 THEN 930
5480 END
5490 GOTO 5490
5500 REM  --MISSILE
5510 CALL SOUND(-90,-5,0)
5520 FOR YI=2 TO 14 STEP 4
5530 CALL HCHAR(YI,X,138)
5540 CALL KEY(0,K,S)
5550 IF S=0 THEN 5690
5560 IF K<>32 THEN 5690
5570 CALL SOUND(-90,4000,0)
5580 CALL VCHAR(1,X,152,14)
5590 CALL VCHAR(1,X,32,14)
5600 CALL HCHAR(YI,X,156)
5610 CALL SOUND(-500,-5,0)
5620 CALL HCHAR(YI,X,32)
5630 SCO=SCO+25
5640 P$=STR$(SCO)
5650 PX=8
5660 PY=21
5670 GOSUB 4920
5680 GOTO 2280
5690 CALL HCHAR(YI,X,32)
5700 NEXT YI
5710 CALL SCREEN(16)
5720 CALL SOUND(-400,-5,0)
5730 CALL SCREEN(2)
5740 GOSUB 5180
5750 GOTO 2280
```

ENDE

CLUB-Info ★ CLUB-Info ★ CLUB-Info

„Zweimal Borken"

Borken/Westfalen (M.K.). Borken (Westf.) scheint ein „gutes Pflaster" zu sein, wenn es um User-Clubs geht. Gleich zwei unterschiedliche „User-Vereinigungen" suchen noch weitere Mitglieder! Alle C-16/116-Freunde werden sich sicher freuen zu hören, daß Thomas Pflewko, Offenbachstraße 15 in 4280 Borken 1, seine Dienste anbietet. Er fragt: „Wollt Ihr einsteigen in unseren Computer-Club? Software und Peripherie sind vorhanden. Monatlich gibt es eine Zeitschrift! Wenn Ihr an uns interessiert seid, schreibt an oben genannte Adresse oder ruft an: Tel: (0 28 61) 6 15 46."

Der „Zweite im Bunde" ist ein Schneider-Club, der von Karsten Kammler „gemanagt" wird. Karsten teilte gegenüber der HOMECOMPUTER-Redaktion mit, daß eine Neugründung vollzogen wurde. Weiter meinte Karsten: „In unserem Schneider-Club kann jeder mitmachen, der im Stadt- oder Kreisgebiet von Borken (westliches Münsterland) wohnt. Ein Club-Beitrag soll nicht erhoben werden. Voraussichtlich wird mindestens einmal im Monat ein Clubtreffen stattfinden. Erfahrungsaustausch, Präsentation von selbstgeschriebenen Programmen und vieles mehr sollen an der Tagesordnung sein.
Schreibt bitte an: Karsten Kemmler, Heinrich-Hertz-Straße 4, 4280 Borken-Gemen. Vergeßt das Rückporto nicht!"

Warum gleich zur Flasche greifen?
Die Computer-Clubs suchen noch Mitglieder!

1490:	20195	1700:	45954	1910:	4176	2110:	57302
1500:	45165	1710:	25257	1920:	29014	2120:	43938
1510:	54213	1720:	1906	1930:	6184	2130:	65502
1520:	44006	1730:	6212	1940:	52067	2140:	58314
1530:	30828	1740:	59734	1950:	48863	2150:	16375
1540:	18848	1750:	5439	1960:	39091	2160:	34436
1550:	37364	1760:	4731	1970:	65436	2170:	38189
1560:	29102	1770:	45866	1980:	19426	2180:	54146
1570:	34818	1780:	44827	1990:	2641	2190:	24116
1580:	53384	1790:	90	2000:	3041	2200:	1852
1590:	18022	1800:	15790	2010:	38121	2210:	32805
1600:	59888	1810:	40542	2020:	52384	2220:	6967
1610:	29219	1820:	50079	2030:	54313	2230:	12668
1620:	57644	1830:	21483	2040:	35403	4990:	0
1630:	24339	1840:	23261	2050:	6447	5000:	59617
1640:	41482	1850:	63445	2060:	23023	5010:	41948
1650:	36229	1860:	41462	2070:	33378	5020:	63836
1660:	667	1870:	35671	2080:	22951	5030:	65006
1670:	0	1880:	51043	2090:	61383	5040:	142
1680:	37873	1890:	56336	2100:	9590		
1690:	42180	1900:	47672				

CRAZY KONG – Gesamtchecksumme: 54133

Ralf, der Clown

TI-99/4A – TI-99/4A – TI-99/4A

Geraten Sie nicht so in Hektik wie dieser Clown auf unserer Abbildung. Sonst könnte Ihnen das gleiche passieren und dann besitzen Sie für ein Engagement im TI-Zirkus keine Chance mehr.

Ihr TI-99-Computer befördert Sie dieses Mal in den Bereich der künstlerisch-creativen Berufe

Der TI-Zirkus benötigt unbedingt einen neuen Clown, da der alte zur besser zahlenden Konkurrenz nach Amerika abgewandert ist. Als Ralf, der zur Zeit ohne Engagement ist, von der freien Stelle „Wind bekommt", macht er sich sofort auf den Weg. Dort angekommen, verlangt der Direktor des Unternehmens natürlich von Ralf eine Solovorstellung.

Beweisen Sie, daß Sie zu einer neuen Attraktion des TI-Zirkus werden können.

Vorhang auf! Zeigen Sie dem Direktor was Sie alles können! Im ersten Bild springen Sie von Trampolin zu Trampolin über die Bühne und müssen springenden Bällen und herumschwirrenden Vögeln ausweichen. Im zweiten Bild balancieren Sie auf Bällen, was einfacher aussieht als es ist. Da so eine Zirkusvorstellung natürlich nicht ewig dauern darf, arbeiten Sie auch gegen die Zeit. Um in das Ensemble aufgenommen zu werden, dürfen Sie höchstens fünf Fehler begehen, aber keinenfalls die Zeit überschreiten. Je schneller Sie Ihr Programm durchziehen können, desto höher wird die Punktezahl (spätere Gage?!) sein.

Einige Tips:

Nach dem Einladen von Kassette und Starten mit „RUN", können Sie eine ausführliche Spielanleitung anwählen. In dieser ist die genaue Handhabung des Spiels erklärt.
Zum Eintippen ist folgendes zu vermerken: Einige Programmzeilen sind länger als fünf Bildschirmzeilen. Sie müssen also unvollständig eingegeben werden, mit „FCTN'REDO" zurückgeholt und dann vervollständigt werden.
Das Spiel ist für die TI-99/4A Konsole mit Extended-Basic und Joystick 1 geschrieben und belegt bis auf wenige Bytes den gesamten Speicher. Besitzer eines Diskettenlaufwerks ohne Speichererweiterung müssen dieses ausschalten, da das Programm sonst nicht läuft.

RALF DER CLOWN — Das Listing

```
100 !***********************
110 !*                     *
120 !*   RALF DER CLOWN    *
130 !*-------------------*
140 !*    (c) 1985 BY      *
150 !* CHRISTIAN ACKERL   *
160 !* STATDPLATZ  22/A   *
170 !* A-4840 VOECKLABRUCK *
180 !*                     *
190 !*-------------------*
200 !*     TI-99/4A &      *
210 !* X-BASIC&JOYSTICK1   *
220 !***********************
230 !
240 !
250 CALL CLEAR :: DISPLAY AT(1,1):"
   RALF DER CLOWN              ---------
-----" :: CALL CHAR(64,"3C4299A1A199423C
")
260 DISPLAY AT(12,1):"  SPIELANLEITUNG (
J/N)? J" :: DISPLAY AT(24,1):" § 1985 BY
 CHRISTIAN ACKERL" :: ACCEPT AT(12,25)SI
ZE(-1)BEEP VALIDATE("JN"):SP$ :: IF SP$=
"" THEN 260
270 IF SP$="N" THEN 440 ELSE CALL CLEAR
:: DISPLAY AT(1,1):"        SPIELANLEITU
NG              --------------"
280 DISPLAY AT(3,1):"das spiel besteht a
us zwei  pruefungen,die ralf bestehenmus
s,damit er den job im    zircus bekommt.
"
290 DISPLAY AT(7,1):"die erste pruefung
besteht  darin,dass ralf einmal zum  rec
hten,dann zurueck zum    linken rand spr
ingt."
300 DISPLAY AT(11,1):"um dieses zu errei
chen,     muessen sie den joystick in di
e gewuenschte richtung    druecken."
310 DISPLAY AT(15,1):"das waere nicht so
 schwer,  wenn da nicht voegel und    sp
ringende baelle waeren,mitdenen man nich
t zusammen-"
320 DISPLAY AT(19,1):"stossen darf.sie m
uessen    auch die vorgegebene zeit   ei
nhalten."
330 DISPLAY AT(24,1):"    WEITER MIT ENT
ER!!" :: CALL KEY(0,K,S):: IF K<>13 THEN
 330 ELSE CALL CLEAR
340 DISPLAY AT(1,1):"ab und zu erscheint
 ueber   den voegeln ein affe,der    ral
f,wenn er ueber ihm ist, und sie den feu
erknopf drue-"
350 DISPLAY AT(5,1):"cken,ein stueck mit
nimmt.dasbringt zusaetzlich punkte.  es
kann aber auch sein,dass er ralf gleich
wieder runterwirft."
360 DISPLAY AT(10,1):"wenn sie ralf wied
er zum    rechten rand gebracht haben,mu
essen sie den joystick    nach links dru
ecken.dann"
370 DISPLAY AT(14,1):"bekommen sie fuer
die uebri-ge zeit einen bonus,und    ko
mmen in das zweite bild.  hier muss ralf
 auf baellen"
380 DISPLAY AT(18,1):"jonglieren und fue
nfmal den ball,auf dem er steht in dervo
rgegebenen zeit wechseln."
390 DISPLAY AT(24,1):"    WEITER MIT ENT
ER!!" :: CALL KEY(0,K,S):: IF K<>13 THEN
 390 ELSE CALL CLEAR
400 DISPLAY AT(1,1):"sie koennen den bal
l auf demralf steht,mit dem joystick hin
 und her lenken,muessen  aber aufpassen,
dass der ball"
410 DISPLAY AT(5,1):"auf dem er steht ni
cht mit  einem anderen ball zusammen-sto
esst.                    wenn sie den ba
ll wechseln"
420 DISPLAY AT(9,1):"wollen,muessen sie
mit dem  joystick in die gewuenschte ric
htung lenken und den     feuerknopf drue
cken."
430 DISPLAY AT(20,1):"  VIEL SPASS ALS C
LOWN !!!" :: DISPLAY AT(24,1):"    WEITE
R MIT ENTER!!" :: CALL KEY(0,K,S):: IF K
<>13 THEN 430 ELSE CALL CLEAR
440 CALL CLEAR :: CALL SCREEN(2):: CALL
MAGNIFY(3)
450 CALL COLOR(0,2,2,1,2,2,2,8,2,8,8,2):
: CALL CHAR(40,"FFBFBFFFFFDFDFFFFFFFFEFE
FCFCF8F8F0F0E0E0C0C080808FFFEF8E080000000
")
460 CALL CHAR(44,"FFFFFFFFFFFEF8E0FFFFFF
7F7F3F1F07FFFFFFFEFEFCF8E0101010387C7C38
10")
470 CALL CHAR(91,"FFFF7F7F3F3F1F1F0F0F07
0703030101FF7F1F0701000000FFFFFFFFFF7F1F
07")
480 CALL CHAR(36,"000000000001073E5EFF00
0000000000000000000E0F81C1FFF")
490 CALL CHAR(60,"000000000001073E5EFE0E
0101000000000000000E0F81C1F1F101010906"
)
500 CALL CHAR(96,"00000000070F1712121310
1010109060000C1E1DFFFCF8080804")
510 CALL CHAR(102,"00003F3F7F7FE3DDBEDDE
BF7EBDDBEFF0000F8F8FCFC8E76FA76AEDEAE76F
AFE")
520 CALL CHAR(106,"000000007F7FFF8000000
000FCFCFE02")
530 CALL CHAR(108,"071F3F7F7FFFFFFFFFFFF
F7F7F3F1F07E0F8FCFEFEFFFFFFFFFFFFFEFEFCF
8E0")
540 CALL CHAR(112,"071F3F7373FEFEFEFEEEE
773783C1F07E0F8FCCECE7F7F7F7F77E7CE1E3CF
8E0")
550 CALL CHAR(116,"010161B31B1B1F1D0F0A0
F070000000000000C9AB0B0F070E0A0E0C000000
000")
560 CALL CHAR(124,"030101010307070703010
30603010B0700604040E070F070E0C0E0B0E0C08
000")
570 CALL CHAR(120,"00060202070E0F0E07030
70D07030100C0808080C0E0E0E0C080C060C080D
0E0")
580 CALL CHAR(128,"000103070D0703070E0F0
E0702020600E0D080C060C080C0E0E0E0C080808
0C0")
590 CALL CHAR(132,"0E1703070D0703070E0F0
E0702020206000030C060C080C0E0E0E0C08080C
000")
600 CALL CHAR(136,"000003070D0703070E0F0
E0702020206000080C060C080C0E0E0E0C080808
```

```
0C0")
610 CALL CHAR(140,"0000100804010210020104081000000000008488904020 9C2040908884000000")
620 CALL COLOR(2,2,2,8,2,2):: CALL CLEAR :: CALL HCHAR(1,1,40,17):: J=1 :: FOR I=15 TO 9 STEP -1
630 CALL HCHAR(J+1,1,40,I-1):: CALL HCHAR(J+1,I,41):: CALL HCHAR(J+2,1,40,I-1):: CALL HCHAR(J+2,I,42):: J=J+2
640 NEXT I :: CALL HCHAR(J+1,1,40,I-2):: CALL HCHAR(J+1,I-1,44):: CALL HCHAR(J+1,I,43)
650 CALL HCHAR(1,18,40,15):: J=1 :: FOR I=15 TO 9 STEP -1 :: CALL HCHAR(J+1,34-I,40,I-1):: CALL HCHAR(J+1,33-I,91)
660 CALL HCHAR(J+2,34-I,40,I-1):: CALL HCHAR(J+2,33-I,92):: J=J+2 :: NEXT I :: CALL HCHAR(J+1,34-I,40,I-1):: CALL HCHAR(J+1,34-I,94)
670 CALL HCHAR(J+1,33-I,93):: CALL HCHAR(2,16,45):: CALL HCHAR(2,17,46)
680 IF AN=0 THEN 690 ELSE CALL SPRITE(#28,116,16,1,121):: MA=5
690 FOR I=1 TO 33 STEP 6 :: CALL HCHAR(23,I,102):: CALL HCHAR(24,I,103):: CALL HCHAR(23,I+1,104):: CALL HCHAR(24,I+1,105):: NEXT I
700 FOR I=1 TO 7 STEP 2 :: CALL HCHAR(17,I,47):: CALL HCHAR(17,I+25,47):: NEXT I
710 DISPLAY AT(1,22)SIZE(4):"TIME" :: TR,YC=1 :: TI=52
720 DISPLAY AT(5,22)SIZE(6):"PUNKTE" :: CALL HCHAR(23,1,106):: CALL HCHAR(23,2,107):: DISPLAY AT(1,3)SIZE(6):"CLOWNS"
730 GOSUB 1210 :: CALL SCREEN(8):: CALL COLOR(13,2,9,9,2,2,10,2,2,8,9,2,2,9,2):: FOR I=3 TO 7 :: CALL COLOR(I,2,9):: NEXT I
740 IF AN=0 THEN 1250
750 CALL SPRITE(#1,136,16,165,1,#2,108,9,176,72,#3,108,5,124,120,#4,108,3,176,168,#5,108,12,128,216)
760 CALL SPRITE(#6,108,14,128,24):: S=10 :: FOR I=7 TO 9 :: CALL SPRITE(#I,36,8,108,S,0,-15):: S=S+70 :: NEXT I
770 RANDOMIZE :: CO=INT(RND*13)+3 :: CALL COLOR(9,CO,2,10,CO,2)
780 FOR I=1 TO 2 :: IF T=176 THEN T=124 :: TT=176 :: PA=108 :: PB=112 ELSE T=176 :: TT=124 :: PA=112 :: PB=108
790 CALL LOCATE(#2,T,72,#3,TT,120,#4,T,168,#5,TT,216,#6,TT,24):: CALL PATTERN(#2,PA,#3,PB,#4,PA,#5,PB,#6,PB):: CALL SOUND(-100,110,15)
800 CALL JOYST(1,Y,X):: IF Y<>0 THEN GOSUB 880
810 NEXT I :: IF RND>.65 AND AF=0 THEN CALL SPRITE(#11,96,9,80,1):: AF=1
820 CALL KEY(1,K,S):: IF K=18 AND AF=1 THEN 1310 ELSE IF VP=36 THEN VP=60 ELSE VP=36
830 FOR I=7 TO 9 :: CALL PATTERN(#I,VP):: NEXT I
840 IF AF=0 THEN 860
850 CALL POSITION(#11,AA,BA):: IF BA=241 THEN CALL DELSPRITE(#11):: AF=0 ELSE CALL LOCATE(#11,AA,BA+48)
860 CALL JOYST(1,Y,X):: IF Y THEN GOSUB 880
870 GOSUB 1380 :: GOTO 770
880 CALL POSITION(#1,A,B):: IF B=241 AND Y=4 OR B=1 AND Y=-4 AND GE=0 THEN RETURN ELSE IF B=193 AND Y=4 THEN GE=1 ELSE IF B=1 AND Y=-4 AND GE=1 THEN 1400
890 CALL SOUND(-1,-3,0):: CALL HCHAR(23,YC,102):: CALL HCHAR(23,YC+1,104)
900 IF Y=-4 THEN TR=TR-1 :: PP=128 ELSE PP=132 :: TR=TR+1
910 CALL PATTERN(#1,PP-8):: CALL LOCATE(#1,A-50,B+Y*6)
920 FOR H=1 TO 10 :: CALL COINC(ALL,C):: IF C THEN 1000
930 NEXT H
940 CALL POSITION(#1,A,B):: CALL LOCATE(#1,A+50,B+Y*6):: CALL PATTERN(#1,PP)
950 IF AN=0 THEN 960 ELSE PU=PU+50 :: GOSUB 1390
960 IF TR=1 THEN YC=1 ELSE IF TR=2 THEN YC=7 ELSE IF TR=3 THEN YC=13 ELSE IF TR=4 THEN YC=19 ELSE IF TR=5 THEN YC=25 ELSE YC=31
970 CALL PATTERN(#1,PP):: CALL SOUND(-1,-2,0):: CALL HCHAR(23,YC,106):: CALL HCHAR(23,YC+1,107)
980 IF FF=1 THEN FF=0 :: GOTO 770
990 RETURN
1000 !
1010 CALL PATTERN(#1,140):: L=0 :: FOR I=660 TO 110 STEP -50 :: CALL SOUND(-100,I,L,I+1,L,I+2,L):: L=L+2 :: NEXT I :: MA=MA-1 :: IF MA=0 THEN 1070
1020 IF BI=1 THEN 1500
1030 IF Y=-4 THEN TR=TR+1 :: LO=24 ELSE IF Y=4 THEN TR=TR-1 :: LO=-24 ELSE LO=0
1040 CALL POSITION(#1,A,B):: CALL LOCATE(#1,165,B+LO):: CALL PATTERN(#1,136):: GOSUB 1210
1050 IF TR=1 THEN YC=1 ELSE IF TR=2 THEN YC=7 ELSE IF TR=3 THEN YC=13 ELSE IF TR=4 THEN YC=19 ELSE IF TR=5 THEN YC=25 ELSE YC=31
1060 CALL DELSPRITE(#11):: BI,AF,FF=0 :: CALL PATTERN(#1,PP):: CALL SOUND(-1,-2,0):: CALL HCHAR(23,YC,106):: CALL HCHAR(23,YC+1,107):: GOTO 820
1070 IF TI>1 THEN 1100
1080 FOR I=1 TO 3 :: DISPLAY AT(3,22)SIZE(5):"O(U(T" :: CALL SOUND(200,659,5):: FOR J=1 TO 100 :: NEXT J :: DISPLAY AT(3,22)SIZE(5):"(((((" 
1090 FOR J=1 TO 100 :: NEXT J :: NEXT I
1100 FOR I=3 TO 13 :: CALL COLOR(I,9,2):: NEXT I :: CALL CLEAR :: CALL DELSPRITE(ALL):: DISPLAY AT(8,1):"      S  C  H  A  D  E "
1110 CALL COLOR(2,2,2,13,2,2):: DISPLAY AT(12,1):"SIE HABEN ES NICHT GESCHAFFT"
1120 BC,BI,FF,TI,MA,GE,PU,AF=0 :: DISPLAY AT(18,1):"   NOCH EIN SPIEL J-N  J"
1130 ACCEPT AT(18,24)SIZE(-1)BEEP VALIDATE("JN"):A$ :: IF A$="J" THEN 540 ELSE IF A$="" THEN 1130
1140 CALL CLEAR :: CALL DELSPRITE(ALL):: END
1150 !
1160 !
1170 FOR I=3 TO 13 :: CALL HCHAR(23,1,32,64):: CALL COLOR(I,9,2):: NEXT I :: CALL CLEAR :: CALL COLOR(2,2,2):: CALL DELSPRITE(ALL)
1180 DISPLAY AT(8,1):"   B    R    A    V    O ": : :"   SIE HABEN DIE PRUEFUNG

 BESTANDEN"
1190 FOR I=110 TO 990 STEP 50 :: CALL SO
```

```
UND(-100,I,5,I+1,7,I+2,9):: NEXT I
1200 GOTO 1120
1210 !
1220 JJ=2 :: FOR I=1 TO MA-1 :: RANDOMIZ
E :: P=INT(RND*2):: IF P=0 THEN P=128 EL
SE P=132
1230 CALL HCHAR(3,JJ,P):: CALL HCHAR(4,J
J,P+1):: CALL HCHAR(3,JJ+1,P+2):: CALL H
CHAR(4,JJ+1,P+3):: JJ=JJ+2 :: NEXT I
1240 CALL HCHAR(3,JJ,40,15-JJ):: CALL HC
HAR(4,JJ,40,14-JJ):: RETURN
1250 CALL COLOR(9,13,2,10,13,2):: CALL S
PRITE(#1,136,16,165,1,#2,116,16,167,241)
1260 S=110 :: FOR I=1 TO 5 :: Y=4 :: GOS
UB 880 :: RANDOMIZE :: CO=INT(RND*13)+3
:: CALL COLOR(9,CO,2,10,CO,2):: CALL SOU
ND(200,S,3,S+1,6,S+2,9):: S=S+111
1270 NEXT I :: CALL PATTERN(#1,136):: L=
110 :: CALL MOTION(#2,-15,-11)
1280 CALL POSITION(#2,A,B):: IF A<20 THE
N CALL DELSPRITE(#2):: CALL SPRITE(#28,1
16,16,1,121)ELSE L=L+50 :: CALL SOUND(-1
00,L,10):: GOTO 1280
1290 S=550 :: FOR I=1 TO 5 :: Y=-4 :: GO
SUB 880 :: RANDOMIZE :: CO=INT(RND*13)+3
 :: CALL COLOR(9,CO,2,10,CO,2):: CALL SO
UND(200,S,3,S+1,6,S+2,9):: S=S-110
1300 NEXT I :: GE=0 :: CALL PATTERN(#1,1
36):: MA=5 :: GOSUB 1210 :: AN=1 :: GOTO
 750
1310 AF,Y=0 :: CALL SOUND(-1,-3,0):: CAL
L HCHAR(23,YC,102):: CALL HCHAR(23,YC+1,
104)
1320 CALL POSITION(#1,A,B,#11,AA,BB):: C
ALL LOCATE(#1,108,B):: CALL COINC(ALL,C)
:: IF C THEN 1010 ELSE IF B<>BB THEN 136
0 ELSE CALL LOCATE(#1,80,B)
1330 CALL POSITION(#11,AA,BA):: IF BA=24
1 THEN 1370 ELSE IF RND>.8 THEN CALL LOC
ATE(#1,165,BA):: CALL DELSPRITE(#11):: F
F=1 :: GOTO 960
1340 PU=PU+200 :: GOSUB 1390 :: CALL LOC
ATE(#11,AA,BA+48,#1,AA,BA+48):: TR=TR+1
:: FOR HH=1 TO 50 :: NEXT HH :: CALL SOU
ND(-100,-7,10)
1350 GOTO 1330
1360 AF=1 :: CALL LOCATE(#1,108,B):: CAL
L COINC(ALL,C):: IF C THEN 1010 ELSE CAL
L LOCATE(#1,165,B):: FF=1 :: GOTO 960
1370 PP=136 :: CALL LOCATE(#1,165,241)::
 CALL DELSPRITE(#11):: GE,FF=1 :: GOTO 9
60
1380 TI=TI-2 :: T$=STR$(TI)&"(" :: DISPL
AY AT(3,23)SIZE(LEN(T$)):T$ :: IF TI=0 T
HEN 1070 ELSE IF BI=1 THEN RETURN ELSE C
ALL PATTERN(#1,136):: RETURN
1390 P$=STR$(PU):: DISPLAY AT(7,23)SIZE(
LEN(P$)):P$ :: RETURN
1400 !
1410 CALL DELSPRITE(ALL):: DISPLAY AT(5,
22)SIZE(6):"BONUS(" :: FOR I=1 TO TI ::
PU=PU+TI :: GOSUB 1390 :: CALL SOUND(-10
0,110+I*40,5):: NEXT I
1420 BC=0 :: BI=1 :: DISPLAY AT(5,22)SIZ
E(6):"PUNKTE" :: CALL HCHAR(23,1,32,64):
: TI=82 :: GOSUB 1380
1430 CALL CHAR(140,"00000003070D07070307
0E0F0E07131F0010189CCC64C8C080C0E0E0E0C0
90F0")
1440 CALL CHAR(128,"070B0103030301030707
0703010101010080C0E0B0E0C0E0707030E0C0C0
C0E0")
1450 CALL CHAR(132,"070B0103030301030707
0703010101010080C0E0B0E0C0E0707070E0C040
20B0")
1460 CALL CHAR(120,"071F3E7E7EFEFFE3E3FF
FE7E7E3E1F07E0F87C7E7EFFFC7C7FF7F7E7E7C
F8E0")
1470 CALL CHAR(124,"071F3F7763F1FBFFFFFB
F163773F1F07E0F8FCEEC68FDFFFFFDF8FC6EEFC
F8E0")
1480 CALL CHAR(116,"000103070D0703070E0E
0E07030204 0DE0D080C0C0C080C0E0E0E0C08080
8080")
1490 CALL MAGNIFY(3):: CALL CHAR(112,"00
0103070D0703070E0E0C0703030307E0D080C0C0
C080C0E0E0E0C080808080")
1500 PP=132 :: GOSUB 1220 :: BA=2 :: Y=1
1 :: FOR I=2 TO 4 :: RANDOMIZE :: CALL S
PRITE(#I,108,12,176,Y):: Y=Y+INT(RND*40)
+70 :: NEXT I :: CALL SPRITE(#1,128,14,1
60,10)
1510 CALL JOYST(1,Y,X):: IF Y=4 THEN PP=
132 ELSE IF Y=-4 THEN PP=116
1520 IF Y=0 THEN CALL MOTION(#1,0,0,#BA,
0,0):: CALL PATTERN(#1,PP-4,#BA,120):: G
OTO 1550 ELSE CALL MOTION(#1,0,Y*2,#BA,0
,Y*2):: CALL PATTERN(#1,PP)
1530 IF BP=120 THEN BP=124 ELSE BP=120
1540 CALL PATTERN(#BA,BP)
1550 CALL POSITION(#BA,A1,B1):: CALL LOC
ATE(#1,160,B1):: IF BD=120 THEN BD=124 E
LSE BD=120
1560 CALL KEY(1,K,S):: IF K=18 THEN 1640
1570 FOR I=2 TO 4 :: IF I=BA THEN 1580 E
LSE CALL MOTION(#I,0,INT(RND*-6)-3):: CA
LL PATTERN(#I,BD):: CALL COINC(#BA,#I,16
,C):: IF C THEN 1690
1580 NEXT I
1590 IF BP=120 THEN BP=124 ELSE BP=120
1600 CALL PATTERN(#BA,BP,#1,PP-4)
1610 CALL KEY(1,K,S):: IF K=18 THEN 1640
 ELSE CALL COINC(#BA,#BA+1,16,C):: IF C
THEN 1690
1620 IF BA>2 THEN CALL COINC(#BA,#BA-1,1
6,C):: IF C THEN 1690
1630 GOSUB 1380 :: GOTO 1510
1640 CALL SOUND(-100,-3,10):: CALL MOTIO
N(#BA,0,-Y*2):: FOR J=-16 TO 16 :: CALL
MOTION(#1,J,Y*2):: NEXT J :: CALL MOTION
(#1,0,0):: CALL POSITION(#1,A,B)
1650 CALL LOCATE(#1,160,B):: FOR I=2 TO
4 :: CALL DISTANCE(#1,#I,C):: IF SQR(C)<
18 AND I=BA THEN BC=BC-1 :: PU=PU-100 ::
 GOTO 1670 ELSE IF SQR(C)<18 THEN 1670
1660 NEXT I :: CALL LOCATE(#1,176,B):: G
OTO 1690
1670 CALL SOUND(-100,-2,10):: BE=I :: CA
LL MOTION(#BE,0,0):: CALL POSITION(#BE,A
1,B1):: CALL LOCATE(#1,160,B1)
1680 GOSUB 1380 :: BA=BE :: PU=PU+100 ::
 BC=BC+1 :: GOSUB 1390 :: IF BC=5 THEN 1
160 ELSE CALL LOCATE(#1,160,B1):: GOTO 1
510
1690 CALL POSITION(#1,A,B):: CALL LOCATE
(#1,176,B):: FOR I=1 TO 4 :: CALL MOTION
(#I,0,0):: NEXT I :: GOTO 1010
```

Listing – The End

Korrektur:

C-64-Programm: „Pool Position"

In der letzten Ausgabe des Homecomputers ist uns leider ein Listing-Fehler unterlaufen. Hier nun die fällige Korrektur:

```
105 DATA,,,,,,,,,,,,,,,,,,6,,,6,,32,15,24
8,120,25,88,156,25,156,132,31,248,4,47     <22>
212 DATA21,208,41,253,141,21,208,141,3,2
08,173,249,7,201,15,208,5,169,7,141,40     <87>
```

„Danger Mission“

Tolles Geschicklichkeitsspiel auf dem TI-99 (Extended Basic)

Du, ich hätte nie geglaubt, daß ich die Bombe noch rechtzeitig entschärfen könnte!“

Es erwartet Sie eine heikle Spielaufgabe!

Beim „Townmarshal“ der Stadt „Blackborn City“ ist eine anonyme Bombendrohung eingegangen. Sie erhalten vom Bürgermeister den gefährlichen Auftrag, das gefährdete Stadtgebiet nach Bomben zu durchsuchen und diese zu entschärfen. Aber aufgepaßt! – Ihr Weg ist mit zahlreichen Gefahren gepflastert! **(S.G.)**

Auf der Suche nach den Bomben werden Sie mit den verschiedensten Bedrohungen konfrontiert. Sie dürfen weder mit den Gebäuden kollidieren, noch von diesen herunterfallen. Bleiben Sie nicht an einer Stelle zu lang stehen, sonst bröckeln die Mauersteine ab und Sie stürzen unweigerlich in den Abgrund. Lange Pausen sind allein deshalb schon nicht gefragt, weil sonst die Bomben explodieren könnten.
Handeln Sie am besten nach dem Motto, erst kurz überlegen, dann agieren.

Steuerung:

Gesteuert wird der „Bombenentschärfer“ bzw. Feuerwerker über Joystick Port 1. Geben Sie zu Beginn die Anzahl der Spieler (1 oder 2) ein. Anschließend bitte den eigenen Namen (3 Buchstaben) eintippen und den gewünschten Level wählen.
Das Programm besteht aus drei verschiedenen Bildschirmmotiven. Falls Sie alle drei Bilder bewältigen sollten, beginnt das Spiel wieder mit dem ersten Bildschirmmotiv, allerdings befinden Sie sich nun in einem schwierigeren Level.

Die verschiedenen Funktionen:

Mit einer senkrechten Stange kommt man in das nächst untere Stockwerk. Man kann die Stange zwar nicht vorher verlassen, jedoch ist es möglich, diese von jeder Position aus zu erreichen. Mit einer Leiter kommt man hoch und runter. Mit einem Lift gelangt man nur nach oben, kann diesen dafür aber jeder Zeit besteigen und verlassen.
Mit einer waagrechten Stange kann man sich nach rechts oder links bewegen. Eine solche Stange erreicht man, indem der Joystick nach oben gedrückt wird; ein Druck auf den Feuerknopf bewirkt, daß man wieder auf die Etage zurückkommt.

```
100 ! DANGER MISSION
110 !
120 ! by REX SOFT
130 !
140 ! written by
150 !
160 ! Daniel Peier
170 ! Hulfteggstrasse 31
180 ! 8400 Winterthur
190 ! Schweiz
200 !
210 ! November-Dezember 1984
220 !
230 DIM A$(5),B$(2),D$(20),P(2),MA(2),L(
2),SCH(2)
240 DATA "   h(,,(    h          %
 h        h,,((,,(x(,         h  %     h
      x          ,,(,',     h        x"
250 GOTO 290 :: A$(),B$(),D$(),N$(),HC$,
CO$,PLAY$,P(),MA(),L(),I,II,SCH(),HC,K,S
,Q,X,Y,X1,X2,Y1,SPIELER,BO,ST,P1,P2,P3,C
A,ANZ
260 CALL CLEAR :: CALL SCREEN :: CALL CO
LOR :: CALL NAME :: CALL SPRUNG :: CALL
SPRITE :: CALL MOTION :: CALL SOUND :: C
ALL LOCATE :: CALL PATTERN
270 CALL GCHAR :: CALL HCHAR :: CALL VCH
AR :: CALL CHAR :: CALL DELSPRITE :: CAL
L POSITION :: CALL KEY :: CALL JOYST ::
CALL COINC
280 !$P-
290 CALL CLEAR
300 CALL CHAR(44,"BBBB00777700DDDD")
310 CALL CHAR(34,"1800583C1A28284818001A
3C58141412",45,RPT$("FF00",4))
320 A$(1)="1C080808083E2A3E" :: A$(2)="0
01C0808083E2A3E" :: A$(3)="0001C08083E2
A3E" :: A$(4)="0000001C083E2A3E" :: A$(5
```

Betätigen Sie sich als Feuerwerker, entschärfen Sie die Bomben und retten Sie die Hausbewohner

```
)="000000001C3E2A3E"
330 CALL CHAR(96,RPT$("818181FF",2),104,
RPT$("81",8),112,"FF",91,"3C42BDA1A1BD42
3C")
340 CALL CHAR(36,"247E99995A3C423C",37,A
$(1))
350 CALL CHAR(128,"5A5A3C181818183C18005
A3C181818241818101E18101018")
360 CALL CHAR(131,"070C1C1F7F92927F00807
FC0FE4949FE78EEFEF8A880847C",134,"824438
5438544482000004 82FF8204")
370 CALL CHAR(63,"7C7EC60C18001818")
380 FOR I=1 TO 15 :: READ D$I):: NEXT I
390 FOR I=65 TO 90 :: READ CO$ :: CALL C
HAR(I,CO$):: NEXT I :: FOR I=48 TO 57 ::
 READ CO$ :: CALL CHAR(I,CO$):: NEXT I
400 CALL SCREEN(8):: CALL COLOR(2,15,1,1
2,16,1,9,14,1,11,5,1,10,7,1,1,13,1,13,11
,1)
410 CALL COLOR(0,1,1,5,9,1,6,9,1,7,9,1,8
,9,1,3,6,1,4,6,1)
420 DISPLAY AT(7,1)ERASE ALL:"D A N G E
R   M I S S I O N" :: CALL HCHAR(5,3,44,
27):: CALL HCHAR(9,3,44,27)
430 DISPLAY AT(24,7):"& REX SOFT 1984"
440 DISPLAY AT(12,1):"PRESS 1 OR 2 PLAYE
R BOTTON 1" :: ACCEPT AT(12,28)SIZE(-1)V
ALIDATE("12"):SPIELER
450 DISPLAY AT(17,1):"PLAYER ONE:   REX
  LEVEL 4" :: ACCEPT AT(17,15)SIZE(-3)VA
LIDATE(UALPHA):N$(1):: ACCEPT AT(17,27)S
IZE(-1)VALIDATE("123456789"):SCH(1)
460 IF SPIELER=2 THEN DISPLAY AT(20,1):"
PLAYER TWO:   DAN   LEVEL 4" :: ACCEPT A
T(20,15)SIZE(-3)VALIDATE(UALPHA):N$(2)::
 ACCEPT AT(20,27)SIZE(-1)VALIDATE("12345
6789"):SCH(2)
470 CALL CLEAR :: MA(1),MA(2)=3 :: Q,ST,
BO,L(1),L(2),P(1),P(2)=0
480 GOTO 1110
490 CALL VCHAR(1,1,32,768):: ON L(Q)+1 G
OSUB 870,940,1020
500 CALL CHAR(120,B$(1)):: CALL NAME(N$(
),SPIELER,P()):: FOR I=L(Q)*5+1 TO L(Q)*
5+5 :: II=II+1 :: DISPLAY AT(2+II*4,1):D
$(I):: NEXT I :: II=0
510 CALL SPRITE(#1,34,7,177,233):: CALL
HCHAR(2,15,32,5):: CALL HCHAR(2,15,130,M
A(Q))
520 CALL POSITION(#1,X1,Y1):: CALL JOYST
(1,X,Y):: CALL KEY(1,K,S):: IF K=18 THEN
 CALL SPRUNG(X1,Y1):: BO=BO-3 :: CALL PO
SITION(#1,X1,Y1):: GOTO 550
530 IF Y THEN 620 :: IF X=-4 THEN CALL P
ATTERN(#1,34):: Y1=Y1-8 ELSE IF X=4 THEN
 CALL PATTERN(#1,35):: Y1=Y1+8
540 CALL LOCATE(#1,X1,Y1):: CALL COINC(A
LL,CA):: IF CA=-1 THEN 830
550 BO=BO-1 :: IF BO<=0 THEN ST=ST+1 ::
CALL CHAR(37,A$(ST)):: BO=100 :: IF ST=5
 THEN 830
560 P1=INT(X1-7)/8+2 :: P2=INT(Y1+1)/8+1
 :: CALL GCHAR(P1,P2,P3)
570 IF P3=37 THEN 600
580 CALL GCHAR(P1+1,P2,P3):: IF P3>39 AN
D P3<43 THEN P3=P3+1 :: CALL HCHAR(P1+1,
P2,P3)ELSE IF P3=43 THEN CALL HCHAR(P1+1
,P2,32)ELSE IF P3=32 THEN 830
590 GOTO 520
600 CALL SOUND(100,110,0,-7,0):: ANZ=ANZ
-1 :: CALL HCHAR(P1,P2,32):: P(Q)=P(Q)+B
O*(6-ST)+L(Q)*50 :: DISPLAY AT(2,5+(Q-1)
*18)SIZE(6):USING "######":P(Q):: IF ANZ
=0 THEN 810
610 GOTO 580
620 P1=INT(X1-7)/8+3 :: P2=INT(Y1+1)/8+1
 :: CALL GCHAR(P1,P2,P3):: IF P3>39 AND
P3<43 THEN P3=P3+1 :: CALL HCHAR(P1,P2,P
3):: IF P3=43 THEN CALL HCHAR(P1,P2,32)
630 CALL COINC(ALL,CA):: IF CA=-1 THEN 8
30 ELSE IF Y=-4 THEN 760
640 CALL GCHAR(P1-1,P2,P3):: IF P3=96 TH
EN CALL LOCATE(#1,X1-8,Y1):: CALL PATTER
N(#1,129):: CALL POSITION(#1,X1,Y1)ELSE
IF P3=104 THEN 710
650 CALL GCHAR(P1-2,P2,P3):: IF P3=112 T
HEN 660 ELSE 560
660 CALL PATTERN(#1,128):: CALL LOCATE(#
1,X1-7,Y1):: CALL POSITION(#1,X1,Y1)
670 CALL JOYST(1,X,Y):: CALL KEY(1,K,S):
: IF K=18 THEN CALL LOCATE(#1,X1+7,Y1)::
 CALL POSITION(#1,X1,Y1):: GOTO 560 ELSE
 CALL COINC(ALL,CA):: IF CA=-1 THEN 830
680 BO=BO-1 :: IF BO<=0 THEN ST=ST+1 ::
CALL CHAR(37,A$(ST)):: BO=100 :: IF ST=5
 THEN 830
690 IF X THEN CALL LOCATE(#1,X1,Y1+X*2):
: CALL SOUND(80,-3,2):: CALL POSITION(#1
,X1,Y1)
700 P1=INT(X1-7)/8+2 :: P2=INT(Y1+1)/8+1
 :: CALL GCHAR(P1,P2,P3):: IF P3=112 THE
N 670 ELSE 830
710 CALL PATTERN(#1,129)
720 CALL POSITION(#1,X1,Y1):: P1=INT(X1-
7)/8+2 :: P2=INT(Y1+1)/8+1 :: CALL GCHAR
(P1,P2,P3):: CALL COINC(ALL,CA):: IF CA=
-1 THEN 830 ELSE IF P3<>104 THEN 520
730 CALL JOYST(1,X,Y):: IF X THEN 520 EL
SE CALL SOUND(100,-6,4):: CALL LOCATE(#1
,X1-8,Y1)
740 BO=BO-1 :: IF BO<=0 THEN ST=ST+1 ::
CALL CHAR(37,A$(ST)):: BO=100 :: IF ST=5
 THEN 830
750 GOTO 720
760 IF P3=96 THEN CALL LOCATE(#1,X1+8,Y1
):: CALL POSITION(#1,X1,Y1):: GOTO 560 E
LSE IF P3<>120 THEN 560
770 CALL PATTERN(#1,129)
780 CALL CHAR(120,B$(2)):: CALL SOUND(10
0,X1+110,3,X1+220,8,X1+500,11)
790 CALL LOCATE(#1,X1+8,Y1):: CALL POSIT
ION(#1,X1,Y1):: BO=BO-1 :: IF BO<=0 THEN
 ST=ST+1 :: CALL CHAR(37,A$(ST)):: BO=10
0 :: IF ST=5 THEN 830
800 CALL CHAR(120,B$(1)):: P1=INT(X1-7)/
8+3 :: P2=INT(Y1+1)/8+1 :: CALL GCHAR(P1
,P2,P3):: IF P3<>120 THEN 560 ELSE 780
810 CALL DELSPRITE(ALL):: L(Q)=L(Q)+1 ::
 P(Q)=P(Q)+BO*(5-ST+SCH(Q))*20 :: ST,BO=
0 :: IF L(Q)>2 THEN SCH(Q)=SCH(Q)+1 :: L
(Q)=1 :: P(Q)=P(Q)+4000+SCH(Q)*1000
820 GOTO 480
830 CALL POSITION(#1,X1,Y1):: CALL PATTE
RN(#1,35):: X2=X2+2 :: CALL MOTION(#1,X2
,0):: IF X1>190 THEN 850
840 CALL PATTERN(#1,34):: CALL SOUND(100
,300-X1,0,400-X1,5,800-X1,9,-7,4):: GOTO
 830
850 CALL DELSPRITE(ALL):: X2,ST,BO=0
860 MA(Q)=MA(Q)-1 :: GOTO 480
870 CALL HCHAR(5,10,37):: ANZ=18
880 B$(1)="2010080404081020" :: B$(2)="0
408102020100804"
890 CALL CHAR(40,"EEEE00777700BBBB8EEA00
3776002BB206E8002574009320064000021640000
12",44,"EEEE00777700BBBB")
900 CALL SPRITE(#5,133,5,177,1,0,SCH(Q),
#6,134,13,57,256,0,-SCH(Q)-3,#7,135,3,13
4,1,0,SCH(Q)+1)
910 IF SCH(Q)>5 THEN CALL SPRITE(#8,132,
13,105,150,0,SCH(Q)-2,#9,131,13,105,142,
0,SCH(Q)-2)
920 IF SCH(Q)>3 THEN CALL SPRITE(#10,133
,7,89,1,0,SCH(Q)+2)
930 RETURN
940 B$(1)="0804081008040810" :: B$(2)="0
810080408100804"
950 CALL CHAR(40,"EEEE00BBBB00DDDDEEEE00
8383009DD9EEC8008183008D59CE00000083000D
40",44,"EEEE00BBBB00DDDD")
```

DANGER MISSION

```
960 ANZ=17 :: CALL HCHAR(5,4,37):: CALL
HCHAR(5,20,37)
970 CALL SPRITE(#5,36,3,33,256,0,-SCH(Q)
-3,#6,133,7,137,1,0,SCH(Q)+2,#7,134,11,6
5,256,0,-SCH(Q)-2)
980 IF SCH(Q)>4 THEN CALL SPRITE(#8,135,
13,105,1,0,SCH(Q)+1)
990 IF SCH(Q)>6 THEN CALL SPRITE(#9,36,2
,177,1,0,SCH(Q)+1)
1000 IF SCH(Q)>7 THEN CALL SPRITE(#10,13
4,15,105,70,0,SCH(Q)+1)
1010 RETURN
1020 B$(1)="1008081010080810" :: B$(2)="
0810100808101008"
1030 CALL CHAR(40,"DDDD00777700BBBBDDDD0
0775300AA09DDDD006542000400DD4500454",44
,"DDDD00777700BBBB")
1040 CALL SPRITE(#5,135,16,177,1,0,SCH(Q
)+2,#6,36,13,153,256,0,-SCH(Q)-3,#7,134,
5,89,256,0,-SCH(Q)-4)
1050 IF SCH(Q)>4 THEN CALL SPRITE(#8,132
,2,105,20,0,SCH(Q)-2,#9,131,2,105,12,0,S
CH(Q)-2)
1060 IF SCH(Q)>6 THEN CALL SPRITE(#10,13
5,9,41,1,0,SCH(Q))
1070 IF SCH(Q)>7 THEN CALL SPRITE(#11,13
5,11,41,50,0,SCH(Q))
1080 IF SCH(Q)>3 THEN CALL SPRITE(#12,36
,14,129,256,0,-SCH(Q)-4)
1090 IF SCH(Q)>8 THEN CALL SPRITE(#13,13
3,15,73,1,0,SCH(Q)+2)
1100 CALL HCHAR(5,27,37):: ANZ=24 :: RET
URN
1110 Q=Q+1 :: IF Q>SPIELER THEN Q=1
1120 IF SPIELER=1 AND MA(1)=0 THEN 1160
1130 IF SPIELER=2 AND MA(1)=0 AND MA(2)=
0 THEN 1160
1140 IF MA(Q)=0 THEN 1110
1150 GOTO 490
1160 DISPLAY AT(9,9)ERASE ALL:",GAME OVE
R," :: CALL HCHAR(8,11,44,11):: CALL HCH
AR(10,11,44,11)
1170 DISPLAY AT(5,7):"DANGER MISSION" ::
 DISPLAY AT(1,10):"REX SOFT"
1180 IF P(1)>HC THEN HC=P(1):: HC$=N$(1)
1190 IF P(2)>HC THEN HC=P(2):: HC$=N$(2)
1200 DISPLAY AT(14,1):USING "HIGHSCORE:
######   BY":HC :: DISPLAY AT(14,26):HC$
1210 DISPLAY AT(18,1):USING "PLAYER ONE:
### ###### POINTS":N$(1),P(1)
1220 IF SPIELER=2 THEN DISPLAY AT(21,1):
USING "PLAYER TWO:### ###### POINTS":N$(
2),P(2)
1230 DISPLAY AT(24,1):"ANOTHER PLAY ???
Y OR N  Y" :: ACCEPT AT(24,26)SIZE(-1)VA
LIDATE("YN"):PLAY$ :: IF PLAY$="Y" THEN
420
1240 FOR I=40 TO 43 :: CALL HCHAR(1,1,I,
768):: NEXT I :: CALL CLEAR :: END
1250 DATA "      '     h       x       p
pppp '     h       x       %      ' %   h
      x       '   (,,((,(,h       x ppp
pp"
1260 DATA "'            h     % x %     '
 %          h    (,(,(,   ,x,((,        h
              x            h    ppppppp
  x"
1270 DATA "   %     %  h    %      %  x(
,'(,,,(,(,,    (('   (,(,(  '
     '          '                  ' %
 %"
1280 DATA "  '                  ,(,(,,(,x
 '  %     %         %   % x(,,((,,,((,(,
((,,((,(,((,,,,"
1290 DATA "x,h(        x((,((,,(((h %  x
 h %       x          h(,(xx ,,((,h    x
          h   xx      h    x          h
  x"
1300 DATA "x      h%   x   h(,((((,((,,(x
     ,((,(,   h          x
   h     ppppppx    pppppppppp h    %
 %"
1310 DATA "x   %     %     h%  h(,   (x(
h(,(,    (   (,,(++h      x h
        h      x h                    h
  x"
1320 DATA " h     %            h %    x
h    x((,((,((,,(x(,(,,h  x h    x
     x     h  x h    x   pppppp  x     h
  x"
1330 DATA " h    x  %      % x %   h  x
h %  x  ,(    (((,(,   h %x,,((,,,
            (,(,"
1340 DATA "%hx pppppppppppppppp  h(((x(
hx%                    h   x h(,(      x
 %     x  h   x h         x ((,h   x  h
  x"
1350 DATA "%h   % %   x    h   x  h   x(
h   ('(   x    h   x  h   x h    '   x
   h % x% h   x h    '    x    ,((((,(h
  x"
1360 DATA " h  % '  % x%              x
h(,((,((((,(x             x%h
x              x,h       %  %x
% x"
1370 DATA " h     x,(h,(x          '(,(
h     x  h  x  pppppp '    h   % x  h
x %       %'    h   (((  h  (,,(  x  (((
"
1380 DATA " h %      h        x        (
(((((h % h %    % x  %           (,(,(,(
(,((,(,((((,((,"
1390 DATA FEFEC6C6FEFEC6C6,FCFEC6FEFEC6F
EFC,FEFECOCOCOCOFEFE,FCFEC6C6C6C6FEFC,FE
FECOF8F8COFEFE
1400 DATA FEFECOF8F8COCOCO,FEFECOCECEC6F
EFE,C6C6C6FEFEC6C6C6,1818181818181818,06
06060606667E3C
1410 DATA C6CCD8FOD8D8CCC6,COCOCOCOCOCOF
EFE,EEEEFED6C6C6C6C6,E6E6E6D6D6CECECE,FE
FEC6C6C6C6FEFE
1420 DATA FCFEC6C6FEFCCOCO,FCFCCCCCCCDCF
CFE,FCFEC6C6FEFCD8CE,FEFECOFEFE06FEFE,7E
7E181818181818
1430 DATA C6C6C6C6C6C6FEFE,6666666666661
818,C6C6C6D6D6D6FEFE,82C66C38386CC682,66
66666618181818,FEFE061C70COFEFE
1440 DATA FEFEE6D6D6CEFEFE,78F8981818181
818,FEFE06FEFECOFEFE,FEFE063E3E06FEFE,CO
COCCFEFEOCOCOC
1450 DATA FEFECOFEFE06FEFE,FEFECOFEFEC6F
EFE,FEFE060C183060CO,FEFEC6FEFEC6FEFE,FE
FEC6FEFE06FEFE
1460 !$P+
1470 SUB SPRUNG(X1,Y1):: GOTO 1480 :: I
:: X :: Y :: X1 :: Y1 :: CALL MOTION ::
CALL PATTERN :: CALL JOYST :: CALL LOCAT
E :: CALL SOUND :: !$P-
1480 CALL JOYST(1,X,Y):: IF X=-4 THEN CA
LL PATTERN(#1,34)ELSE IF X=4 THEN CALL P
ATTERN(#1,35)
1490 FOR I=1 TO 3 :: CALL MOTION(#1,-10,
1.5*X):: CALL SOUND(100,110+I*30,2):: NE
XT I :: FOR I=3 TO 1 STEP -1 :: CALL MOT
ION(#1,10,2*X):: CALL SOUND(100,110+I*30
,0):: NEXT I
1500 CALL MOTION(#1,0,0):: X=MAX(MIN(Y1+
4*X,256),1):: CALL LOCATE(#1,X1,X)
1510 !$P+
1520 SUBEND
1530 SUB NAME(N$(),SPIELER,P()):: GOTO 1
540 :: N$():: SPIELER :: P():: CALL HCHA
R :: !$P-
1540 CALL HCHAR(1,3,45,28):: CALL HCHAR(
3,3,45,28)
1550 DISPLAY AT(2,1):N$(1):: DISPLAY AT(
2,5)SIZE(6):USING "######":P(1)
1560 IF SPIELER=2 THEN DISPLAY AT(2,19):
N$(2):: DISPLAY AT(2,23)SIZE(6):USING "#
#####":P(2)
1570 !$P+
1580 SUBEND
```

The House Of Horror

Sie sind frischgebackener Erbe eines schmucken Hauses! Leider hat das Ganze einen dicken Haken. In der Zeit nämlich, in der das Haus leergestand, hat sich ein eierköpfiger Hexenmeister mit seinen Geistern in das Haus einquartiert. Nun müssen Sie diese ungebetenen Mieter erst einmal vertreiben, um Ihr Erbe unbeschwert genießen zu können.

Für den TI 99 4 A (auch ohne X-Basic!!!)

Spielbeschreibung:

Aber ganz alleine schaffen Sie das nicht. Zu Hilfe eilt eine Fee, die in den Räumen Zauberkräfte und Wunderkugeln verteilt hat, die Sie einsammeln müssen. Diese Zauberdinge wirken vernichtend auf die Geister des Hexenmeisters.
Bei Spielbeginn befindet sich unser junger Hausbesitzer in der untersten Etage auf der linken Seite. Jetzt heißt es, alle Hindernisse schadlos zu überstehen, um nach oben zu dem Hexenmeister zu gelangen und ihm mit den unterwegs eingesammelten Wunderkugeln kräftig einheizen. Allerdings haben die Wunderkugeln keine Wirkung, wenn man zu nahe an den gefährlichen Objekten steht. Außerdem: die Aufgänge zur 3. und 4. Etage öffnen sich erst, wenn 2 Geheimnisse gelöst wurden!

Steuerung:

Die Knollennase des Helden zeigt immer die Laufrichtung an.
Taste S : Schritt nach links
Taste D: Schritt nach rechts
Taste J : Sprung in Blickrichtung
Taste T : Zauberkugel in Blickrichtung werfen

In den Etagen lauern vielfältige Gefahren! Kleine dicke Gespenster tauchen wie zufällig auf und wollen Ihnen den Weg versperren. In der 2. Etage gibt es zudem noch Stalaktiten, die herunterfallen können. In der 3. Etage erwartet Sie ein unsichtbarer Geist und in der 4. Etage empfängt Sie der Zauberlehrling und der Hexenmeister wirft mit Kegeln um sich.

Tips zur Lösung:

So oft wie möglich, sollten Hindernisse natürlich übersprungen werden. Der unsichtbare Geist kann mit Wunderkugeln abgeschossen oder durch Springen überlistet werden. Man kann auch warten, bis er durch umherirrende Geister oder herunterfallende Steine zerstört worden ist. Der Hüpfgeist in der 4. Etage sollte sofort mit Wunderkugeln traktiert werden. Wenn man zu wenig Wunderkugeln besitzt, kann man die löchrige Wand in der Mitte auch niederlaufen. Ist das geschafft, muß man dem Zaubermeister eine Zauberkugel genau auf den rechten Fuß werfen. Sollte dies ebenfalls gelingen, können Sie sich als Sieger beglückwünschen lassen.

TI/99

```
100 REM ******************          * SI
MPLE SOFTWARE *          *    presents
  *          ******************
110 REM * HOUSE OF HORROR *          ****
**************          *for ti & ex-bas
ic*          *   C 1984 by     *
*  HANS STOLICKA  *
120 REM ******************
130 FOR I=1 TO 12
140 CALL COLOR(I,16,1)
150 NEXT I
160 CALL SCREEN(2)
170 CALL CLEAR
180 PRINT "     ****************
    *SIMPLE SOFTWARE*          ********
********* ":::::::
190 PRINT "  simple software proudly
                                    pres
ents"::::
200 CALL SOUND(250,262,0,330,0,392,0)
210 CALL SOUND(250,262,0,330,0,392,0)
220 CALL SOUND(250,440,0,523,0)
230 CALL SOUND(250,440,0,523,0)
240 CALL SOUND(550,523,0,659,0)
```

TI/99

Das Listing von „HOUSE OF HORROR"

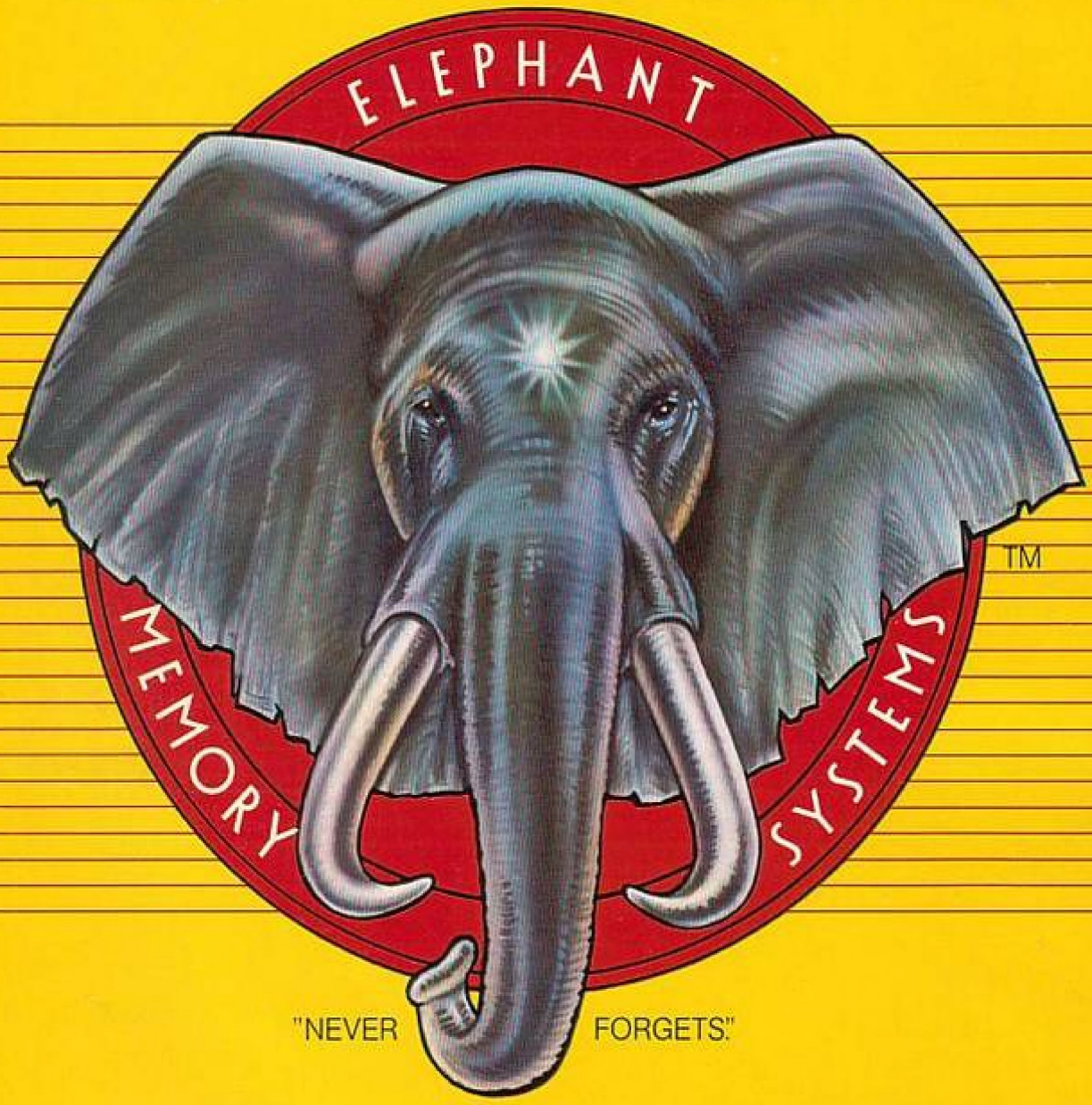
```
250 FOR I=1 TO 500
260 NEXT I
270 CALL CLEAR
280 DIM RITTE$(8,2)
290 RITTE$(1,1)="H.STOLICKA"
300 RITTE$(1,2)="RANG 2"
310 RANDOMIZE
320 CALL CHAR(106,"1838083C5A18246C")
330 CALL CHAR(126,"00000000183C3C18")
340 CALL CHAR(107,"18380AFC191FA040")
350 CALL CHAR(96,"00FBFBFB00DFDFDF")
360 CALL CHAR(131,"00183C18183C3C18")
370 CALL CHAR(112,"FF7E7C3C383010")
380 CALL CHAR(108,"181C103C5A182436")
390 CALL CHAR(105,"003C7E5A7E665A7E")
400 CALL CHAR(120,"0000000000183C18")
410 CALL CHAR(111,"003C7E5A7E3C243C")
420 CALL CHAR(119,"FFDDFFBBFFED5544")
430 CALL CHAR(118,"F8D0F8A0F0E05040")
440 CALL CHAR(95,"427E4242427E4242")
450 CALL CHAR(89,"00070A1F180F021F")
460 CALL CHAR(59,"0248600BA000280A")
470 CALL CHAR(60,"00C1024005810608")
480 CALL CHAR(90,"000080CACC9C0AF0")
490 CALL CHAR(122,"3C5A7E24187EA566")
500 CALL CHAR(88,"3C5A3C187EBD24E7")
510 CALL CHAR(92,"80808000B0C080E0")
520 CALL CHAR(93,"3C5A3C187EBD24E7")
530 CALL CHAR(91,"2FAF6FA70D180D3D")
540 CALL CHAR(98,"00EE007700EE0077")
550 CALL CHAR(136,"000080C0C08000C0")
560 CALL CHAR(137,"E0928C0E9AC080E0")
570 CALL COLOR(8,4,1)
580 CALL COLOR(9,9,16)
590 CALL COLOR(10,16,1)
600 CALL COLOR(11,11,1)
610 CALL COLOR(12,6,1)
620 CALL COLOR(13,10,1)
630 CALL COLOR(14,8,1)
640 FOR I=2 TO 7
650 CALL COLOR(I,16,1)
660 NEXT I
670 CALL COLOR(1,2,1)
680 CALL CLEAR
690 CALL SCREEN(2)
700 PRINT "oooooooooooooooooooooooooooo"
::::"  T H E"::::"-----------------------
-----oooooooooooooooooooooooooooo"
710 PRINT "----------------------------"
::::::::"   C 1984  HANS STOLICKA "::"ooo
oooooooooooooooooooooooooo"::
720 P$="H O U S E  O F  H O R R O R "
730 TY=12
740 TX=3
750 GOSUB 1840
760 FOR I=1 TO 400 STEP 10
770 CALL SOUND(-350,110,0,147,3,I+110,0,
-4,0)
780 NEXT I
790 CALL CLEAR
800 INPUT "SCHWIERIGKEITSGRAD(1-7)? ":F
810 IF (F<1)+(F>7)THEN 800
820 IF F<6 THEN 840
830 PRINT ::"DU BIST WOHL ANFAENGER,WAS
?"
840 PRINT ::::::::::::::::::::::::::::::
:
850 REM  INTRO ENDE & START
860 VERSUCH=1
870 X=21
880 Y=3
890 CALL CLEAR
900 D=INT(RND*20)+6
910 N=F*5+5
920 ZK=5
930 REM  SPIELFELD
940 PRINT ''''''''''''''''''''''''''''''
wwww'wwwww'www'ww'wwww'www''     w      '
  w  w    '   ''
950 PRINT "'  YZ       w            w   ''
8888      w          o    ' '''''''''''''''
''''''''''''''''
960 PRINT "' pppp' pppppp ' pppp'pp'ww''
     p        '    p  p  ''
  p            ''
970 PRINT "'    o     o          '  o  ''
''''''''''''''''''''''''''''' ppppp'ppppp
'p'ppppp'ppppp'"
980 PRINT "'       '      p p      '     ''
     '    -        '     ''X xo  b
     o          ''
990 PRINT ''''''''''''''''''''''''''''' _''
wwww'p'ppppp'pp'p'wwwww''_''     p '
'  ' p     p'_'"
1000 PRINT "        p      p  b          _'
   o       o     b  o x  o _''''''''' '''
 ''''''''' '''''"
1010 PRINT ''''''''''''''''''''''''''''''
"
1020 TX=3
1030 TY=1
1040 P$="WUNDERKUGELN : "
1050 GOSUB 1840
1060 FOR I=1 TO 3
1070 CALL HCHAR(16,INT(RND*14)+11,93)
1080 NEXT I
1090 CALL HCHAR(6,INT(RND*10)+15,120)
1100 CALL HCHAR(1,19,42,ZK)
1110 CALL HCHAR(11,INT(RND*20)+7,120)
1120 CALL HCHAR(11,INT(RND*20)+7,120)
1130 CALL HCHAR(11,D,35)
1140 CALL HCHAR(16,INT(RND*15)+12,120)
1150 CALL HCHAR(16,INT(RND*15)+12,120)
1160 CALL HCHAR(1,28,86)
1170 CALL HCHAR(1,29,58)
1180 CALL HCHAR(1,30,VERSUCH+48)
1190 TX=3
1200 TY=24
1210 P$="KEHRE LIEBER GLEICH UM   "
1220 GOSUB 1840
1230 CALL HCHAR(11,29,131)
1240 OP=INT(RND*10)+15
1250 CALL HCHAR(6,OP,122)
1260 CALL HCHAR(X,Y,108)
1270 NIX=0
1280 CALL KEY(0,R,T)
1290 IF T=0 THEN 1280
1300 REM  START ACTION
1310 UN=INT(RND*N)+1
1320 IF UN=4 THEN 3100
1330 CALL GCHAR(6,7,G)
1340 IF G=32 THEN 4100
1350 REM  STEUERUNG (AENDERUNG FUER JOYS
TICK HIER)
1360 CALL KEY(0,R,T)
1370 IF R=74 THEN 1580
1380 IF R=83 THEN 1500
1390 IF R=68 THEN 1420
1400 IF R=84 THEN 2570
1410 GOTO 1310
1420 CALL HCHAR(X,Y,32)
1430 Y=Y+1
1440 CALL GCHAR(X,Y,G)
1450 IF G<>32 THEN 1900
1460 CALL GCHAR(X+1,Y,G)
1470 IF G=32 THEN 1900
1480 CALL HCHAR(X,Y,108)
1490 GOTO 1310
1500 CALL HCHAR(X,Y,32)
1510 Y=Y-1
1520 CALL GCHAR(X,Y,G)
1530 IF G<>32 THEN 1900
1540 CALL GCHAR(X+1,Y,G)
1550 IF G=32 THEN 1900
1560 CALL HCHAR(X,Y,106)
1570 GOTO 1310
1580 CALL GCHAR(X,Y,U)
1590 CALL HCHAR(X,Y,32)
1600 IF U=108 THEN 1720
1610 CALL GCHAR(X-1,Y-1,G)
1620 IF G<>32 THEN 1900
1630 Y=Y-2
```

```
1640 CALL HCHAR(X-1,Y+1,107)
1650 CALL GCHAR(X,Y,G)
1660 CALL HCHAR(X-1,Y+1,32)
1670 IF G<>32 THEN 1900
1680 CALL GCHAR(X+1,Y,G)
1690 IF G=32 THEN 1900
1700 CALL HCHAR(X,Y,106)
1710 GOTO 1310
1720 CALL HCHAR(X,Y,32)
1730 CALL GCHAR(X-1,Y+1,G)
1740 IF G<>32 THEN 1900
1750 Y=Y+2
1760 CALL HCHAR(X-1,Y-1,107)
1770 CALL GCHAR(X,Y,G)
1780 CALL HCHAR(X-1,Y-1,32)
1790 IF G<>32 THEN 1900
1800 CALL GCHAR(X+1,Y,G)
1810 IF G=32 THEN 1900
1820 CALL HCHAR(X,Y,108)
1830 GOTO 1310
1840 REM  STRING AN BELIEB. STELLE DRUCK
EN
1850 FOR STR=1 TO LEN(P$)
1860 CALL HCHAR(TY,(TX+STR)-1,ASC(SEG$(P
$,STR,1)))
1870 NEXT STR
1880 RETURN
1900 REM   ABPRUEFEN HINDERNISS
1910 IF G<>120 THEN 1990
1920 CALL SOUND(110,980,0)
1930 CALL SOUND(100,765,0)
1940 ZK=ZK+1
1950 CALL HCHAR(1,19,32,9)
1960 CALL HCHAR(1,19,42,ZK)
1970 CALL HCHAR(X,Y,108)
1980 GOTO 1310
1990 IF (G<>88)*(G<>95)THEN 2200
2000 CALL HCHAR(X,Y,107)
2010 FOR I=X-1 TO X-5 STEP -1
2020 CALL SOUND(-100,210,0)
2030 CALL HCHAR(I,Y,107)
2040 CALL HCHAR(I+1,Y,95)
2050 FOR P=1 TO 10
2060 NEXT P
2070 CALL HCHAR(I,Y,107)
2080 NEXT I
2090 X=X-5
2100 N=N-2
2110 IF G=131 THEN 2150
2120 IF G=95 THEN 2190
2130 P$="VORSICHT UNSICHTBARES ETWAS"
2140 GOTO 2160
2150 P$="GUT GEMACHT                "
2160 TX=3
2170 TY=24
2180 GOSUB 1840
2190 GOTO 1310
2200 IF G<>131 THEN 2240
2210 CALL GCHAR(X,D,TR)
2220 IF TR<>35 THEN 2000
2230 GOTO 3850
2240 IF G<>119 THEN 2260
2250 GOTO 2270
2260 IF G<>118 THEN 2460
2270 AG=INT(RND*3)
2280 CALL GCHAR(6,14,G)
2290 IF G=118 THEN 2400
2300 CALL SOUND(120,-6,0)
2310 CALL HCHAR(6,14,118)
2320 IF AG<>0 THEN 2370
2330 CALL SOUND(100,766,0)
2340 P$="GENICK GEBROCHEN           "
2350 GOSUB 1840
2360 GOTO 3850
2370 Y=Y+1
2380 CALL HCHAR(X,Y,106)
2390 GOTO 1310
2400 CALL SOUND(120,-6,0)
2410 CALL HCHAR(6,14,32)
2420 IF AG=0 THEN 2330
2430 Y=Y+1
2440 CALL HCHAR(X,Y,106)
2450 GOTO 1310
2460 IF G<>35 THEN 2520
2470 FOR I=1 TO 6
2480 CALL HCHAR(X,D,111)
2490 CALL HCHAR(X,D,32)
2500 NEXT I
2510 GOTO 3850
2520 IF G<>32 THEN 2550
2530 X=X+1
2540 CALL HCHAR(X,Y,107)
2550 GOTO 3850
2570 REM  ZAUBERKUGEL WERFEN
2580 IF ZK=0 THEN 3020
2590 ZK=ZK-1
2600 CALL HCHAR(1,19,32,9)
2610 CALL HCHAR(1,19,42,ZK)
2620 CALL GCHAR(X,Y,G)
2630 IF G=106 THEN 2790
2640 CALL GCHAR(X,Y+1,G)
2650 IF G<>32 THEN 2940
2660 CALL HCHAR(X,Y+1,120)
2670 CALL GCHAR(X,Y+2,G)
2680 CALL HCHAR(X,Y+1,32)
2690 IF G=96 THEN 2940
2700 IF G=119 THEN 3070
2710 CALL SOUND(630,-7,0)
2720 FOR I=1 TO 6
2730 FOR K=59 TO 60
2740 CALL HCHAR(X,Y+2,K)
2750 NEXT K
2760 NEXT I
2770 CALL HCHAR(X,Y+2,32)
2780 GOTO 1310
2790 CALL GCHAR(X,Y-1,G)
2800 IF G<>32 THEN 2940
2810 CALL HCHAR(X,Y-1,120)
2820 CALL GCHAR(X,Y-2,G)
2830 CALL HCHAR(X,Y-1,32)
2840 IF G=96 THEN 2940
2850 IF G=119 THEN 3070
2860 CALL SOUND(600,-7,0)
2870 FOR I=1 TO 6
2880 FOR EX=59 TO 60
2890 CALL HCHAR(X,Y-2,EX)
2900 NEXT EX
2910 NEXT I
2920 CALL HCHAR(X,Y-2,32)
2930 GOTO 1310
2940 CALL SOUND(100,1834,0)
2950 CALL SOUND(100,3500,0)
2960 P$=" KUGEL VERSCHWENDET !!     "
2970 TY=24
2980 TX=3
2990 GOSUB 1850
3000 GOTO 1310
3010 TY=24
3020 P$=" * KEINE KUGEL DA *        "
3030 TX=3
3040 TY=24
3050 GOSUB 1850
3060 GOTO 1330
3070 REM  MAUER ANGESCHOSSEN
3080 CALL SOUND(100,-5,0)
3090 CALL HCHAR(6,14,118)
3100 REM   !! UNHEIL !!
3110 ON (X-1)/5 GOTO 3120,3450,3450,3670
3120 REM  UNHEIL 4.ETAGE
3130 CALL GCHAR(X,OP,G)
3140 IF G<>122 THEN 3290
3150 SX=INT(RND*13)+15
3160 CALL HCHAR(X,OP,32)
3170 CALL GCHAR(X,SX,G)
3180 FOR I=1 TO 5
3190 CALL HCHAR(X,SX,G)
3200 CALL HCHAR(X,SX,122)
3210 NEXT I
3220 IF G<>32 THEN 3850
3230 CALL SOUND(100,278,0)
3240 OP=SX
3250 N=N-2
```

```
3260 IF N<6 THEN 3270 ELSE 3280
3270 N=6
3280 GOTO 1330
3290 REM   ETAGE 4 - 2.TEIL
3300 IF Y<14 THEN 1330
3310 N=INT(6+(F/1.6))
3320 TIP=INT(RND*5)+8
3330 CALL HCHAR(5,7,136)
3340 CALL HCHAR(6,7,137)
3350 FOR I=8 TO TIP
3360 CALL SOUND(100,193,0)
3370 CALL GCHAR(6,I,G)
3380 CALL HCHAR(6,I,126)
3390 IF G<>32 THEN 3850
3400 CALL HCHAR(6,I,32)
3410 NEXT I
3420 CALL HCHAR(5,7,90)
3430 CALL HCHAR(6,7,92)
3440 GOTO 1330
3450 REM   UNHEIL 2.+ 3.ETAGE
3460 WQ=INT(RND*25)+4
3470 CALL GCHAR(X-3,WQ,G)
3480 IF NIX>14 THEN 3670
3490 IF G<>112 THEN 3460
3500 NIX=NIX+1
3510 CALL SOUND(390,150,0)
3520 CALL HCHAR(X-3,WQ,32)
3530 CALL HCHAR(X-2,WQ,112)
3540 CALL HCHAR(X-1,WQ,112)
3550 CALL HCHAR(X-2,WQ,32)
3560 CALL GCHAR(X,WQ,G)
3570 CALL HCHAR(X-1,WQ,32)
3580 IF Y=WQ THEN 3850
3590 FOR I=1 TO 5
3600 FOR EX=59 TO 60
3610 CALL HCHAR(X,WQ,EX)
3620 NEXT EX
3630 NEXT I
3640 CALL HCHAR(X,WQ,32)
3650 HK=INT(RND*2)+1
3660 IF HK=1 THEN 1330
3670 REM  UNHEIL 1.ETAGE
3680 CALL SOUND(200,220,0)
3690 CV=INT(RND*22)+6
3700 CALL GCHAR(X,CV,Q)
3710 CALL HCHAR(X,CV,105)
3720 IF (Q=106)+(Q=108)THEN 3850
3730 CALL GCHAR(X+1,CV,G)
3740 IF G<>32 THEN 3840
3750 CALL HCHAR(X,CV,32)
3760 CALL HCHAR(X+1,CV,105)
3770 CALL SOUND(650,-7,0)
3780 FOR I=1 TO 5
3790 FOR EX=59 TO 60
3800 CALL HCHAR(X+1,CV,EX)
3810 NEXT EX
3820 NEXT I
3830 CALL HCHAR(X+1,CV,32)
3840 GOTO 1330
3850 REM  HELD TOT
3860 CALL SOUND(1000,-7,2)
3870 FOR EX=1 TO 10
3880 FOR I=59 TO 60
3890 CALL HCHAR(X,Y,I)
3900 NEXT I
3910 NEXT EX
3920 CALL HCHAR(X,Y,32)
3930 P$="PECH GEHABT
3940 GOSUB 1840
3950 FOR I=1 TO 100
3960 NEXT I
3970 P$=" ?? NEUER VERSUCH (J/N) ??"
3980 GOSUB 1840
3990 CALL KEY(0,R,T)
4000 IF T=0 THEN 3990
4010 IF R=78 THEN 4050
4020 VERSUCH=VERSUCH+1
4030 IF VERSUCH>9 THEN 4050
4040 GOTO 870
4050 CALL CLEAR
4060 PRINT " ICH GLAUBE DU BIST DIESER
                                AUFGABE NI
CHT GEWACHSEN"
4070 PRINT ::::::::::
4080 PRINT "NEUSTART MIT RUN !"
4090 END
4100 REM  AUFGABE ERFUELLT
4110 CALL SOUND(-3000,-7,0)
4120 FOR I=1 TO 5
4130 FOR EX=59 TO 60
4140 CALL HCHAR(6,7,EX)
4150 CALL HCHAR(5,6,EX)
4160 CALL HCHAR(6,6,EX)
4170 CALL HCHAR(5,7,EX)
4180 NEXT EX
4190 NEXT I
4200 CALL SOUND(-2000,-7,0)
4210 FOR I=1 TO 20
4220 CALL SCREEN(16)
4230 CALL SCREEN(9)
4240 CALL SCREEN(2)
4250 NEXT I
4260 CALL SCREEN(3)
4270 CALL CLEAR
4280 RESTORE
4290 PRINT "   ***********************
  *                    *     *********
************** ":
4300 FOR I=1 TO 10
4310 READ ZZ,SS
4320 CALL HCHAR(21,8+WW,ZZ)
4330 WW=WW+2
4340 TY=21
4350 CALL SOUND(300,SS,0,SS+2,3,SS-2,3)
4360 NEXT I
4370 KL=INT(F+(VERSUCH/3-1))
4380 PRINT :"** G U T E   A R B E I T  *
*----------------------------"
4390 PRINT "DU HAST ALLE GEISTER IM
HOUSE OF HORROR ERLEDIGT !! ------------
----------------"
4400 PRINT "WEGEN DEINER UEBERRAGENDEN
LEISTUNGEN IN DER MODERNEN  GESPENSTERBE
KAEMPFUNG "
4410 PRINT "ERHAELTST DU DEN GESPENSTER-
ORDEN ";KL;". KLASSE. ":"---------------
-------------"
4420 INPUT "NAME ? ":NR$
4430 PRINT "VIELEN DANK. "
4440 FOR I=1 TO 100
4450 NEXT I
4460 FOR I=1 TO 8
4470 IF RITTE$(I,1)="" THEN 4490
4480 NEXT I
4490 RITTE$(I,1)=NR$
4500 RITTE$(I,2)="RANG "&STR$(KL)
4510 CALL CLEAR
4520 PRINT :"LISTE DER ORDENSTRAEGER :
 oooooooooooooooooooooooooooo":
4530 FOR I=1 TO 8
4540 PRINT RITTE$(I,1),RITTE$(I,2)
4550 NEXT I
4560 PRINT ::"oooooooooooooooooooooooooo
oo                          NEUES SPIE
L (J/N) ?"
4570 CALL KEY(0,R,T)
4580 IF T=0 THEN 4570
4590 RANDOMIZE
4600 CALL SCREEN(2)
4610 IF R=74 THEN 270
4620 DATA 71,247,82,262,65,262,84,494,85
,494,76,392,73,392,69,330,82,330,69,3000
0
```

Das der TI-99 immer noch voller positiver Überraschungen ist, beweist nicht zuletzt dieses Programm!

DOPPELT GEWINNEN MIT "ELEPHANT"!



Jetzt gibt es neben dem 1. Gewinn, der überragenden Qualität der "ELEPHANT"-Floppy-Disk, noch einen triftigen Grund mehr, schnell Ihren "ELEPHANT"-Händler aufzusuchen: ein Puzzle-Spiel, das Ihnen einen schönen Extra-Gewinn einbringen kann. Was es zu gewinnen gibt, steht auf dem fertigen Puzzle!
Also nichts wie hin zu Ihrem "ELEPHANT"-Spezialisten.

Dennison

ELEPHANT™ NEVER FORGETS.

MARCOM Computerzubehör GmbH

This collection is for all TI99 user.
If you like Ti99 computer join at international forum present on site AtariAge.
Greetings from Italy. Astharot